Final Report

Phase III

DEVELOPMENT OF TECHNIQUES FOR IMPROVING THE RELIABILITY

OF DIGITAL SYSTEMS THROUGH LOGICAL REDUNDANCY

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA

CONTRACT M-44501 UNDER
NASA CONTRACT NASw-6

By: Jack Goldberg   J. A. Baer   R. C. Minnick

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA   *SRI

August 1963

Final Report

Phase III

DEVELOPMENT OF TECHNIQUES FOR IMPROVING THE RELIABILITY

OF DIGITAL SYSTEMS THROUGH LOGICAL REDUNDANCY

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA

CONTRACT M-44501 UNDER
NASA CONTRACT NASw-6

By: Jack Goldberg J. A. Baer R. C. Minnick

S R I Project No. 3196

Approved:

DAVID R. BROWN, MANAGER   COMPUTER TECHNIQUES LABORATORY

JERRE D. NOE, DIRECTOR   ENGINEERING SCIENCES DIVISION

Copy No. 11

ABSTRACT

This is the final report on the third year of a study of techniques for improving the reliability of digital systems through the application of redundancy.

The objectives of this year's research were (1) continuation of the previous year's study of the design of reliable memory read access switches and extension to over-all memory system considerations, (2) to study means for correcting faults in the amplifiers (known as transfer circuits) of the Inhibit-Core computer scheme by means of logical redundancy, and (3) to develop a breadboard version of a reliable Inhibit-Core Transfer Circuit employing circuit redundancy.

In the study of reliable memory techniques, consideration is given to the design of recoders for reliable read access switches. The function of such recoders is to convert the code used in the external computer system into the set of signals needed by the power drivers, which in turn drive a fault-masking read access switch. Recoders are developed for non-zero-noise access switches, and for Constantine, Latin Square, and Sobelman access switches. Different circuit techniques are presented, including the use of single-aperture magnetic cores, with and without diodes, and multi-aperture cores. Consideration is also given to the design of magnetic code-correction circuits that are based on block codes, for the correction of errors in the information channels.

In the study of logical redundancy for Inhibit-Core Logic amplifier protection, it is assumed that the logic elements are perfectly reliable— i.e., that all faults occur in the amplifiers. Since these have only a transmission and detection function, the methods of error-correcting coding for communication channels are applied. The problem is to find codes and circuits for the encoding and decoding functions which have a reasonable practical realization in the framework of a standard Inhibit-Core system. The report describes a number of feasible schemes which offer the system designer variable trade-offs among redundancy costs— i.e., in encoding, decoding, and amplifier equipments, and in time.

iii

3

In the development of a breadboard version of a reliable transfer circuit, the major concern is failure of the semiconductor components by short or open circuit conditions among the terminals, in any combination. A circuit was developed and built that meets the client's performance requirements over all specified temperature and power supply conditions in the presence of any single transistor or diode failure, and in the presence of a number of multiple failures. A number of interesting alternative circuit approaches are also described.

FOREWORD

For the past three years, Stanford Research Institute has conducted a study aimed at developing techniques for improving the reliability of digital systems through the application of redundancy. This study has been supported by the Jet Propulsion Laboratory because of its interest in the construction of digital equipment which is required to operate in space, unattended, for long periods of time.

Although there are many published theoretical analyses of the performance of redundant systems, the amount of practical engineering techniques available for the design of redundant circuits and systems is very limited. Furthermore, the well-known methods—series paralleling at the component level, and majority functions at the logical level are quite expensive. They are "universal" methods, in that they may be applied to any part, but there are important parts of a digital system which may be corrected at significantly lower cost (important costs in this situation are electrical and material loading of the supporting system, and the increased likelihood of parts failure due to the increased number of parts and to the increased complexity of the equipment). In addition, component redundancy is difficult to apply to a circuit without lowering its operating margins.

The objective of the work of the past three years has been to develop techniques for protecting different sections of digital systems against faults, at low cost. It was found that this economy could be achieved by taking advantage of the special characteristics of the different sections, and by employing combinations of components in which the more reliable components provide protection for the less reliable components. The major topics studied in the first two years were (1) the automatic detection of faults in combinational switching networks,[1] (2) the design of codes and coding circuitry for fault correction in digital systems,[2] (3) the application of redundancy for the design of reliable magnetic core memories,[3] and (4) the design of simple stored-program computers

for the efficient application of redundancy.[3] These studies were theoretical and general, although several practical questions arose in the memory work.

The work of the third year was more practical in nature. In addition to having general long-range interests in redundancy techniques, the client is currently engaged in the design of a digital computer system based on the inhibit-core logic scheme, a well-known circuit scheme, characterized by a high ratio of magnetic to semiconductor components and low standby power consumption. The first and third sections of this final report present the results of studies of two approaches to the application of redundancy to the inhibit-core scheme— i.e., circuit redundancy, and logical redundancy.

In the first study, the sense amplifier (Transfer Circuit) was examined, because it contains most of the semiconductors in the system, and these are considered to be the components most likely to fail. A breadboard circuit was developed using the performance specifications of the non-redundant system currently being developed by the client. This circuit meets these specifications under all simulated single-fault conditions, and a number of multiple-fault conditions. Many alternative circuit configurations were examined and rejected in the development of the final circuit.

The second study also concentrated on the sense amplifiers, but it was based on the application of error-correcting code techniques, in which a group of amplifiers are combined in such a way that the desired signals are produced even under complete transient or permanent failure of amplifiers—one or more, depending on the complexity of the code. A number of techniques were developed which require the use of somewhat more complex forms of the standard output circuit (a magnetic-core diode current-steering switch). The different techniques call for different proportions of redundancy among input-logic, amplifier, and output-logic equipment, giving the system designer some flexibility of choice. Generally, the redundancy ratios are low.

The two approaches to amplifier protection may be used individually or in combination. Inasmuch as the problems are extracted from a whole system, it is difficult to recommend a "best" course. The choice must be made by a system designer with the over-all view of systems compatibility,

of equipment boundary conditions, of component costs, and of the degree
of reliability improvement required.

The second major study of the third year, presented in Part Two,
was the continuation of the previous year's work on reliable magnetic-
core memories. The main subject was the design of recoder switches,
whose function is to translate the code employed in the external computer
system to the set of signals required for energizing the drivers of
magnetic-core access switches. Several kinds of access switches were
studied in the previous year. These are capable of correcting faults
in the drivers, so that this study completed the development of efficient
means for accessing a memory with amplifiers subject to faults. Magnetic
circuits were also developed for efficient correction of errors in the
information channels, using error-correcting codes. This study, although
not related to a particular memory development, has aimed at development
of practical techniques.

The studies of the past three years have produced many redundancy
methods, which allow a design to take advantage of special circumstances
of circuit function and of the relative costs in reliability and material
of different available component types. These techniques were developed
separately and in general terms. It is suggested that they be tested
by application to a complete working system.

7

CONTENTS

CONTENTS

9

ILLUSTRATIONS

10

TABLES

PART ONE: DESIGN OF A RELIABLE TRANSFER CIRCUIT (CIRCUIT SENSE AMPLIFIER) USING CIRCUIT AND COMPONENT REDUNDANCY

By

J. A. BAER

I INTRODUCTION

The primary objective of this portion of the project was to develop and test a breadboard version of a special form of sense amplifier, called a "transfer circuit," that achieves high reliability through semiconductor redundancy and through the use of magnetic devices. The transfer circuit is to be compatible with the client's realization of an Inhibit-Core computer. A secondary objective was to evolve circuits and techniques that are applicable to the development of a reliable transfer circuit.

Both of these objectives have been achieved. A reliable circuit has been developed and tested in the form of a laboratory breadboard. It maintains full operating margins even with semiconductor faults present. Redundancy provides protection against all modes of failure of any single semiconductor component and additionally protects against certain multiple failures. A major part of the effort has been directed toward developing and implementing semiconductor redundancy techniques, although some magnetic circuitry has also been investigated. Magnetics has really come into the work largely through the liberal use of ferrite pulse transformers.

II NON-REDUNDANT TRANSFER CIRCUIT

In an Inhibit-Core logic system[4] the transfer circuit performs the
functions of signal detection, noise rejection, pulse amplification,
pulse shaping, and storage of information. A particular implementation
of a transfer circuit that is of interest to the client is shown in
Fig. 1.1. This circuit was taken as the starting point for this portion
of the project. The transfer circuit proper consists of a sense amplifier
and a blocking oscillator that perform the functions of signal detection,
pulse amplification, and a part of the noise-rejection function. Noise
is rejected by virtue of the voltage threshold of the base-emitter junc-
tion of the silicon transistors, and by the finite energy required for
triggering the blocking oscillator. The strobe circuit enclosed in the
dashed line is not really a part of the transfer circuit. It performs
the remaining part of the noise discrimination function by essentially
turning the amplifier stage on and off. The amplifier is active only
at READ time (*i.e.*, at Logic READ and at Register READ); signals that
occur at other times are not amplified. The current-steering switch[5]
performs the storage and pulse-shaping function. It is significant to
note that the blocking-oscillator output pulse, when present, overrides
the pulse from the READ clock in setting the current-steering switch.

FIG. 1.1 NON-REDUNDANT TRANSFER CIRCUIT

III REDUNDANCY TECHNIQUES

Redundancy techniques can be considered as belonging to one of two categories—namely, signal redundancy and circuit redundancy. In signal redundancy, information is added to the basic "message"; in circuit redundancy hardware is replicated. It is circuit redundancy that was applied in this portion of the work. Circuit redundancy can be divided into circuit-block redundancy and component redundancy. The circuit block that is replicated could consist of a very few components or of a large number of components forming a module of some sort. By component redundancy we mean replication on a resistor, transistor, etc., level. The common approaches to circuit redundancy make use of von Neumann[6] or majority logic redundancy, and Moore-Shannon[7] or quad redundancy, although there are other useful approaches.[8,9]

It is a difficult task to fully assess the reliability increase that redundancy affords in any specific instance. One can get a meaningful evaluation by applying two figures of merit, one of which is an indication of the price that is being paid for reliability and one that is an indication of the increase in reliability that has been achieved. The former is the ratio of the number of components used in the redundant circuit to the number of components used in the non-redundant circuit (with each circuit performing the same electrical function). This figure of merit is called the *redundancy ratio*, or more correctly, the *component redundancy ratio*.

The figure of merit that is a measure of the increased reliability can take several forms. One way to establish the figure of merit is to compare the "mean time to failure" of the redundant and non-redundant circuits.[10] Another way is to compare the two circuits from a probability of success (or failure) standpoint, knowing numerically the probability of success or failure of the individual components. Since it is difficult to obtain reliability data on individual components that can be accurately applied to a specific application one cannot ordinarily make accurate predictions. Nevertheless, meaningful estimates can be made.[11] When the numerical probability is unknown, one can establish a ratio between

17

the reliability in the redundant and non-redundant cases, and express
this ratio as a function of the individual component reliabilities, which
are assumed to be the same in both circuits. Calculations of this type
are made in Sec. IV-E of Part One.

IV   REDUNDANT TRANSFER CIRCUIT

A.   GENERAL

In this section we will discuss a particular redundant circuit.
This circuit, in the form of a laboratory breadboard model, is one of
the tangible results of this portion of the project. A major goal was
to evolve such a redundant circuit and carry it through the breadboard
stage.

B.   SCOPE

This circuit evolved assuming that semiconductor failures are the
chief cause of transfer circuit failure, and that semiconductor redundancy
is to be used to protect against these failures. Failures are assumed to
be of a catastrophic nature, and it is acceptable, if undesirable, for
the transfer circuit to momentarily malfunction during the occurrence of
a component failure. It is further assumed that the probability of a
component failing by shorting is equal to the probability of the component
failing by opening. PNP transistors, zener diodes, four-layer diodes,
silicon controlled rectifiers, and backward diodes are considered to be
undesirable components for this application since it has not been demon-
strated that they are highly reliable components. On the other hand,
the NPN transistors and the diodes used in the non-redundant circuit,
Fig. 1.1, are considered to be acceptable components.

The clock-current pulse and the strobe voltage are assumed to behave
in an ideal fashion. Furthermore, the following are assumed to be much
more reliable than semiconductors:

> (1)   Wire and insulation

> (2)   Connections

> (3)   Capacitors

> (4)   Resistors

> (5)   Chokes

> (6)   Pulse transformers

> (7)   Square-loop toroids.

19

The operational constraints that are placed on the circuit are as follows:

(1) The strobe is to activate the circuit at READ time but it is not to be used for noise discrimination during the READ-pulse time interval.

(2) The standby power of the circuit is to be low (<50 Mw).

(3) The clock repetition rate, number of clock phases, and the shape and amplitude of the clock pulse all have fixed values.

(4) The input signal source impedance, the amplitude and shape of the signal pulse, and the variations of these parameters with temperature are all fixed.

(5) The output characteristics of the redundant circuit are to be essentially the same as the characteristics of the present current-steering switch.

(6) The circuit is to operate properly throughout the ambient temperature range of -10°C to +85°C, with a simultaneous variation of ±50% and ±20% in power supply voltages as indicated in Fig. 1.1.

C. A TWO-CHANNEL CIRCUIT

In Fig. 1.2 is shown a block diagram representation of the original non-redundant circuit, omitting the current-steering switch. Each rectangle represents one transistor and some associated parts. It is instructive to ask what would happen if circuit-block redundancy is applied in a manner that would give two separate signal channels. Such a circuit is indicated in Fig. 1.3. If one transistor in either channel opens up, then the remaining channel is relied upon to perform the circuit function. If the input and output circuits of the good channel are not disturbed and if one output is sufficient to drive the load—*i.e.*, the
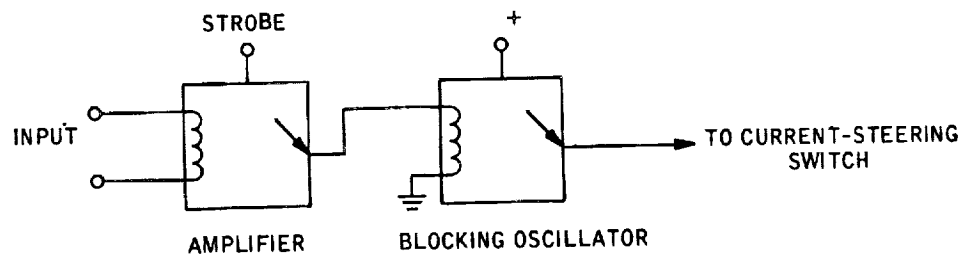
FIG. 1.2 BLOCK DIAGRAM OF THE NON-REDUNDANT CIRCUIT

STROBE   +

INPUT   STROBE   +   TO SWITCH

AMPLIFIERS   BLOCKING-OSCILLATORS

RA-3196-76
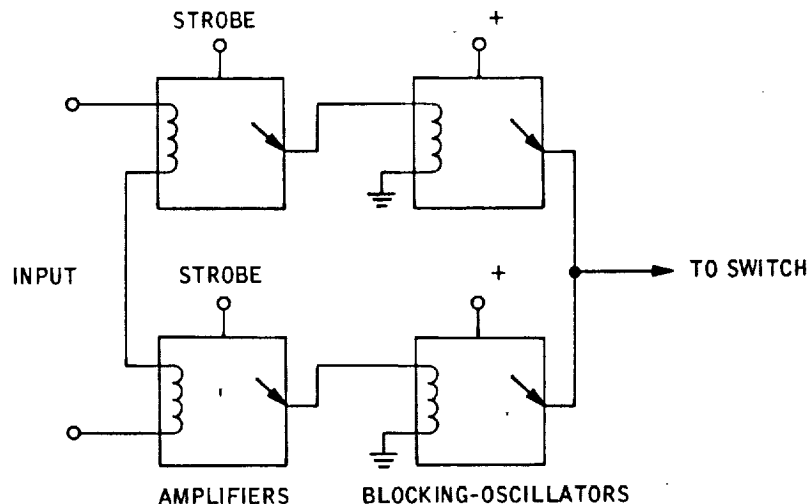
FIG. 1.3 BLOCK DIAGRAM OF A TWO-CHANNEL REDUNDANT CIRCUIT

current-steering switch, then the circuit will indeed protect against this failure. But consider the effect of one transistor shorting from collector to base. If this short occurs in a blocking oscillator stage, then dc is applied to the output terminal at all times, and the current-steering switch will be incorrectly set (see Fig. 1.1). Therefore, the circuit does not protect against a false *1* arising in this manner. If the short occurs in the transistor in an amplifier stage, then the strobe voltage is applied directly to the input of the blocking oscillator and an output pulse occurs every time the strobe voltage appears. Again a false *1* is generated.

The paralleling of circuit blocks has protected against a false *0* but has not protected against a false *1*. However, if we were to put two transistors in series in their collector-to-emitter circuits, then a short in a single transistor would be protected against. A circuit block consisting of two series transistors in an amplifier stage and two series transistors in a blocking oscillator stage could be placed in parallel with an identical circuit block, and the entire unit would in a sense be an acceptable one. There is a practical difficulty that arises from such a circuit for this application. Namely, the sense winding now supplies a signal to four transistors instead of one. The power available at each of the four transistors is less than that available to the one transistor in the non-redundant circuit, and it turns

13

out in this application that the four cannot be driven hard enough to develop sufficient trigger voltage for the blocking oscillator. Therefore, this circuit cannot be used here.

Before proceeding further let us summarize the steps taken so far. First we found that the simple two-channel circuit would not protect against a false 1; note that this circuit has a transistor redundancy ratio of 2. Secondly, we found that by increasing the transistor redundancy ratio to 4 we could protect against both a false 1 and a false 0, but that for this application the limited power available from the input did not permit proper circuit operation even in a fault-free condition.

The next step we take increases the transistor redundancy ratio to 5, and we can provide the additional power needed to drive four transistors; such a circuit is illustrated in Fig. 1.4. Here the sense winding is connected to two transistors; each of these in turn drives two more transistors. The first amplifier stages are connected to dc rather than the strobe voltage, so series transistors are not needed. The
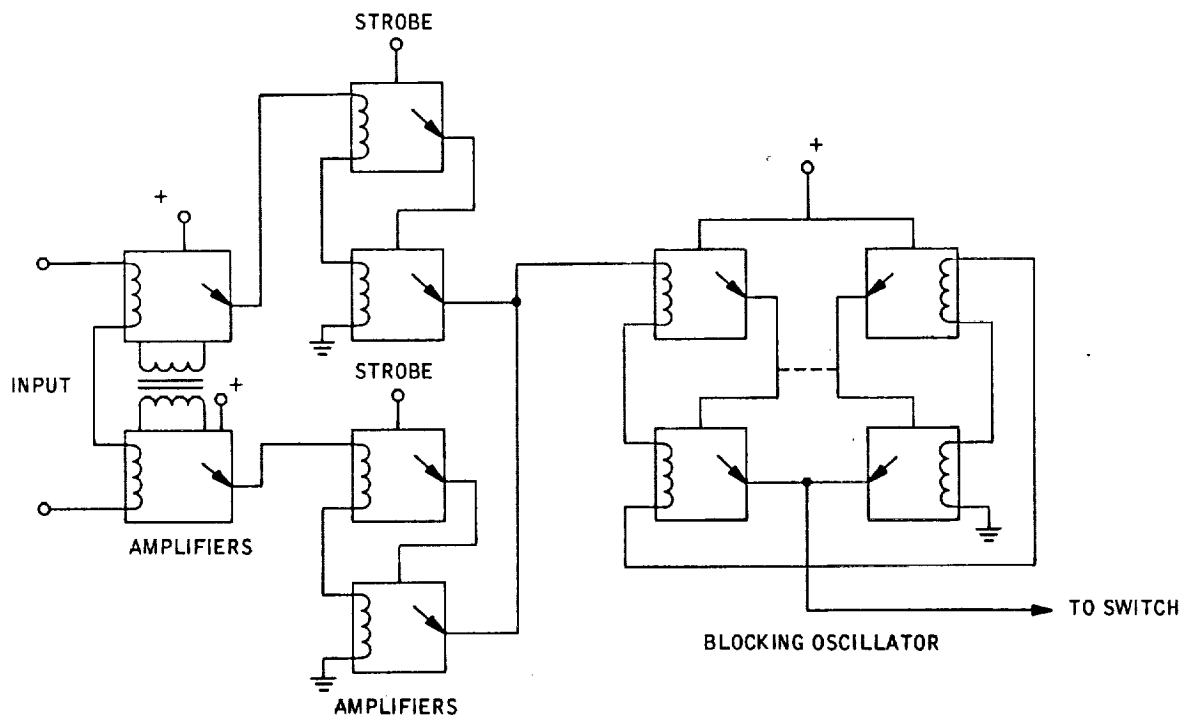
FIG. 1.4 BLOCK DIAGRAM OF THE FINAL REDUNDANT CIRCUIT

14

transformer connecting these amplifiers is for amplitude stabilization, as will be explained in the next section. The transfer circuit as a whole is activated at the proper time by applying the strobe voltage to the second amplifier stages where transistors are put in series for fault protection. The outputs of the second amplifier stages could each go to series transistors with each series pair forming a blocking oscillator stage. The redundant circuit would then comprise two separate channels. However, the reliability of the transfer circuit can be increased by connecting the amplifier outputs in parallel to drive the quadded blocking oscillator as indicated in Fig. 1.4. The dashed line that connects the quad at its midpoint indicates an optional conducting path. If the probability of a transistor failing by opening is greater than the probability of its failing by shorting, then the midpoints should be connected. For the reverse situation, the midpoints should not be connected. A circuit having its midpoints joined is generally the more difficult one to implement.

D. EVOLVED CIRCUIT

The block diagram circuit shown in Fig. 1.4 represents the circuit which was evolved in this portion of the work, although the steps taken did not follow the ones indicated above. The complete schematic of this redundant circuit, except for the current-steering switch, is shown in Fig. 1.5. The current-steering switch that is used with this circuit is the same as the original switch shown as a part of Fig. 1.1, except that the three diodes are replaced by three diode quads.

In the circuit shown in Fig. 1.5 each collector circuit has a resistor in it to limit the current drawn from the power supply in the event a transistor shorts. There are additional resistors in the collector circuit of the blocking oscillator and second amplifier stages. These resistors permit the individual transistors to saturate even though a particular transistor may have a different value for its saturation voltage than does the transistor in parallel with it. The series transistors also have resistors shunting them (collector to emitter) to equalize the collector-to-emitter voltage when the transistors are not saturated.

The quadded blocking oscillator circuit is essentially the original circuit repeated four times. The trigger windings of all the transformers

23

are connected in series, as are the collector windings. It is interesting to note that four transformers are used here when a cursory examination might indicate that one transformer should be sufficient. Four are required in order to maintain the operating margins of the original circuit when the redundant circuit has faults in it. That this is indeed the case can be appreciated by considering a quad circuit having only one transformer but having four secondary (base) windings. For this circuit, if one transistor failed by shorting from base to collector, approximately one-half of the load current would flow through the associated base winding and the short. The other half of the load current would flow through the parallel set of transistors which is not faulty. The current through the base winding is detrimental to the circuit operation because the output pulse width is determined in part by the saturation characteristics of the transformer core. The load current flowing in the base winding moves the operating point of the core and therefore changes the width of the pulse. This change is reflected as a decrease in the range of supply voltages over which the circuit will operate.

Putting the transistors in series in the second amplifier stage is straightforward except for the diode quad shunting the base circuit winding. The purpose of this quad differs from the typical purpose of a diode in this position. It is not required here as long as there are no faults in the circuit. When a short occurs between the collector and the base of one of the series transistors, a weak burst of oscillations occurs each time the strobe voltage appears. The feedback path for these oscillations is through the first amplifier stage in some manner—probably through the transistor capacitance. The diodes across the base winding prevent the occurrence of these fault-induced oscillations. (It might be possible to reduce the number of diodes required by putting the diodes across the series primary windings. This was not tried.)

The first amplifier stages have several features that are required to maintain operating margins in the presence of faults. Three features which stabilize the amplifier output amplitude will now be described. In the absence of the capacitor in the base circuit, a base-to-collector short of a transistor in one channel permits dc to flow through the secondary of the input transformer associated with that channel. This current flowing through eighty turns saturates the transformer core;

16

FIG. 1.5 REDUNDANT TRANSFER CIRCUIT SCHEMATIC DIAGRAM

ALL PULSE TRANSFORMERS ARE
WOUND ON EMI CORE # 143-100

ALL DIODES ARE 1N658

this effectively places a short across the transformer input terminals, and the signal presented to the other channel is considerably increased. The signal increase *per se* is not a detriment, but the discrimination against noise becomes difficult, hence the capacitor.

The inductor shown as a dotted line across the transformer input terminals represents the magnetizing inductance of the transformer. In the usual pulse transformer design this inductance is purposely made high to minimize any shunting effect. In this application the inductance is made low (about 6 μh) to provide a shunt path for the signal in the event that a transistor opens between its base and emitter. In the absence of such a shunt path the signal to the other channel would have its amplitude considerably reduced.

The transformer that provides coupling between the two base circuits stabilizes the signal amplitude in the event that a transistor shorts between its base and emitter. In the absence of this transformer a base-emitter short would be reflected as a short across the input terminals of the transformer feeding this transistor. As in the case of the saturated transformer, too great a noise signal would be present at the input of the other channel. The transformer couples degeneratively into each base circuit from the signal in the other base circuit. A base-emitter short in one channel causes a large degenerative signal to be coupled into the good channel, counteracting the amplitude increase present at its input terminals.

With the exception of the 2N915 transistor used in the first amplifier stages, the semiconductor types used in the redundant circuit are the same as those in the original non-redundant circuit. Component evaluation was not an intended part of this work. The 2N915 is used because the 2N1613, 2N1711, and the 2N2297 were found to be unsuitable for this amplifier stage, although further circuit work might lead to a method of using them if this were considered desirable. (Conversations with personnel from Fairchild Semiconductor Co. reveal that the 2N915 is probably as reliable as the 2N1613 for this particular application.) It is the collector-to-base capacitance that causes trouble with the 2N1613, etc.; degeneration via this capacitance reduces the stage gain below a desirable level. The 2N915 has a much lower capacitance than does the 2N1613 (3.5 pf as opposed to 25 pf) and is therefore a good transistor for this amplifier circuit.

26

The 2N1613's and the 2N2297's as used in the redundant circuit should be more reliable than they are individually in the non-redundant circuit. This comes about because they are subjected to less electrical stress. By putting transistors in series, the collector-to-emitter voltage is reduced by a factor of two when the transistors are active and the circuit is operating without faults. Both the second amplifier stages and the blocking oscillator have transistors in series. The blocking oscillator has an additional feature which should further increase the reliability of the individual transistors. The load current is shared by two parallel paths so each transistor is required to carry only one-half of the current the single blocking oscillator transistor carries in the non-redundant circuit.

Note that no tolerance values are given on any of the components in the circuit. No explicit effort was made to determine the limits that are permissible; however, experience with the circuit indicates that none of the component parameters is critical except for the matching indicated for the first amplifier input transformers.

Table 1.1 is a summary of the properties of the redundant circuit shown in Fig. 1.5.

E. EVALUATION OF THE REDUNDANT CIRCUIT

A breadboard of the circuit shown in Fig. 1.5 was subjected to a series of tests over the temperature range of -10°C to +85°C with supply voltage variations of ±20% and ±50%. The input signal for these tests came from two different sources that simulate the actual sources in the computer. This simulation was specified by the client. One source simulates a group of Register cores and comprises a string of twenty, 30-50 ferrite toroids, Electronic Memories, Inc., type 56-102, spaced such that the entire string was three inches in length. A two-turn sense winding linked the cores and connected to ten inches of a twisted pair of #26 teflon insulated wires. One core only was switched at READ time.* The READ clock pulse has an amplitude of 600 milliamperes, a width at the base of the pulse of 2 microseconds, a rise time of 0.5 microseconds, and a fall time of 0.2 microseconds. This pulse drives the Register cores through a two-turn winding.

_____

*Much of the testing was done with a simulation that is now obsolete because coincident-current operation was used. The client changed from this type of operation during the project to one where only a single core at any one READ time is subjected to a drive pulse, thereby considerably reducing the noise problem.

Table 1.1

SUMMARY OF CIRCUIT CHARACTERISTICS

FIRST AMPLIFIER STAGE	
Purpose: gain, threshold	
Major Features	Functions
Two parallel channels	Protects against false 0
"Cross-coupling" transformer	Stabilizes signal level under certain fault conditions
"Poorly" designed input transformer	Stabilizes signal level under certain fault conditions
Coupling capacitor	Stabilizes signal level under certain fault conditions
Minor Features	Functions
Collector resistor	Protects power supply
Emitter resistor	Stabilizes gain vs. temperature, time and manufacturing tolerance
Transformer output coupling	Converts transistor short into a 0
SECOND AMPLIFIER STAGE	
Purpose: gain, false signal discrimination	
Major Features	Functions
Two channels with outputs combined	Protects against false 0
Series transistors	Protects against false 1
Diode quad	Suppresses oscillations when collector-base shorts
Minor Features	Functions
Resistive divider	Equalizes V_{CE}
Common collector resistor	Maintains amplitude of output when one channel fails. Protects power supply
BLOCKING OSCILLATOR	
Purpose: gain, threshold, pulse stretching	
Major Features	Functions
Transistor quad	Protects against false 0 and 1
Separate transformers	Maintains operating margins
Minor Features	Functions
Resistive divider	Equalizes V_{CE}
10-ohm collector resistors	Equalizes collector currents
Common collector resistor	Limits pulse current amplitude. Maintains current amplitude when transistors fail
Diode quad	Protects base-emitter against high reverse voltage
Series base resistor	Influences output pulse width
Parallel RC	Suppresses noise

The other source simulates a group of Logic cores and comprises a string of forty-six 50-80 ferrite toroids, Ampex Computer Products, type 802-40, spaced such that the entire string was six inches in length. A one-turn sense winding linked the cores and connected to fifteen inches of a twisted pair of #26 teflon insulated wire. Here only one core at a time is in the state that will permit read-out, and the entire string is driven simultaneously by the READ clock pulse. Noise cancellation cores are also in the sense loop and are driven by the READ clock. The clock pulse goes through one turn on each Logic core.

The load used for these tests was the current-steering switch shown in Fig. 1.1.

The circuit was found to operate satisfactorily with no fault, with any single fault, and with certain combinations of multiple faults. A fault in a transistor is considered to be an open or a short between any two or three terminals in all combinations. Transistor failures were simulated by shorting between the socket terminals with clip leads and by opening the circuit at the socket. For the most part this was done with the transistor removed from the socket, but some simulated faults were put in with the transistor *in situ*. Because of the very large number of combinations of multiple faults that are possible for the entire circuit when one takes into account the eleven modes of failure* of a single transistor, only selected combinations were tested. These combinations were chosen to give the most severe circuit loading. An example of a combination that is protected against is a collector-to-emitter short in a single transistor in each of the three stages (considering for this purpose that the first amplifier stage consists of two transistors and the second amplifier stage consists of four transistors). As an extreme, if one can select the modes and specific transistors for failure, it is possible for seven of the ten transistors to fail and the circuit still operates with full margins. The circuit operated satisfactorily with all combinations of faults that were tested (when the combination was one presumably protected against). More exhaustive testing would be a desirable part of any further work on this circuit.

---

*
For any given transistor type, some of the combined short-open failure modes will be very unlikely. The eleven modes of failure are: three single-terminal open-circuit conditions, three two-terminal shorts, three single-terminal-open-two-terminal-short combinations, one fully opened circuit (given by any two- or three-terminal open circuits), and one fully shorted circuit (given by any two pairs of short circuits).

The failures against which this circuit is primarily intended to give protection are catastrophic in nature. What is considered most important here is the effect that the failure has upon the circuit after it has occurred, rather than its effect during the occurrence of the failure. It should be noted, however, that this redundant circuit protects against a decrease in the gain of a transistor whether it occurs suddenly or over a long period of time, and protection is in fact afforded during the occurrence of a failure in most cases. One transient failure that is not completely protected against is a short in the first amplifier stage. If such a short happens to occur during the time that the strobe voltage is applied to the second amplifier stage, then a false 1 could be generated. The fact that the strobe has a duty factor of about 1:5 leads to a low probability that this transient condition will cause trouble.

To determine how much reliability improvement the redundant circuit gives, a probability analysis can be made. To do this in great detail is a formidable problem. One would need to know the probability of occurrence of each mode of failure for each particular component under the actual environmental and electrical conditions the circuit will experience. This kind of information is, of course, not available. It is nevertheless possible to get meaningful information by making calculations based upon certain reasonable assumptions.

First consider the probability of success of the non-redundant circuit shown in Fig. 1.1. It contains seven diodes and two transistors. The reliability of a diode and a transistor are assumed to be roughly the same. Denoting this reliability of the individual semiconductor (or probability of success) by "p," the probability of success of the entire circuit can be expressed as p^9. (Note that we are taking into account only semiconductor failures.) Since in this circuit it is trivial, from an operational point of view, to replace all the diodes by diode quads, it is instructive to compare the reliability of the completely non-redundant circuit with a circuit susceptible only to transistor failures (two transistors). For the latter, the probability of success is p^2, which is a big improvement. Now consider the probability of success of the redundant circuit shown in Fig. 1.5. This analysis becomes more involved than these above. An initial step can be taken by assuming that the circuit fails if there are two component failures. This is a pessimistic view, since there are many ways that two and more components

30

can fail without causing circuit failure. The probability of failure
for the redundant circuit in this case is approximately $45q^2$, for small
q, where q is $(1 - p)$, the probability of semiconductor failure. A
second step can be taken if one knows or assumes a value for the relative
probability of occurrence of a failure by shorting compared to the failure
by opening. It appears reasonable to assume an equal probability of
occurrence of shorts and opens. On the basis of this assumption the
probability of failure of the redundant circuit becomes approximately
$5.5q^2$, which is a considerable improvement over the non-redundant circuit.

$$\text{The ratio:} \quad \frac{\text{probability of failure of the non-redundant circuit}}{\text{probability of failure of the redundant circuit}}$$

is a useful figure of merit. When redundancy is applied on a component
level to the diodes only, this ratio is approximately 4.5. When complete
semiconductor redundancy is applied and shorts and opens assumed equally
likely, the ratio becomes approximately $1/0.6q$.

It is instructive to pursue this probability evaluation further.
As an illustration of the amount of improvement which might result, let
us assume that these three types of transfer circuits are to be used in
a computer requiring 30 such circuits and having a mission time of 10^4
hours, or about 1¼ years. We will also assume that the semiconductor
mean-time-to-failure (MTF) lies between 2 million hours and 20 million
hours.[11, 12] For these assumptions the probability of failure of one
semiconductor, q, becomes 0.005 for 2 million hours MTF and 0.0005 for
20 million hours MTF. The probability of success of the mission is
given in the following table along with other data. Where the redundant
circuit of Fig. 1.5 is involved it is assumed that failure by shorts
and opens are equally probable.

From Table 1.2 it is apparent that semiconductor redundancy can be
applied to gain substantial improvement in the transfer circuit reliability,
bearing in mind the assumptions that have been made, and also bearing in
mind that the computer must have an extremely high probability of success
when it is but one of many sub-systems in the entire space vehicle.

Table 1.2

CIRCUIT RELIABILITY COMPARISON

CHARACTERISTIC	NON-REDUNDANT CIRCUIT (Fig. 1.1)	WITH QUADDED DIODES	FULLY REDUNDANT CIRCUIT (Fig. 1.5)
Transistor redundancy ratio	1	1	5
Diode redundancy ratio	1	4	6.3
Probability of *failure* for one circuit	$9q$	$2q$	$5.5q^{2}$*
Probability of *failure* for one circuit with 2 million hours MTF assumed*	0.045	0.01	1.4×10^{-4}
Probability of *failure* for one circuit with 20 million hours MTF assumed*	0.0045	0.001	1.4×10^{-6}
Probability of computer *success* with 2 million hours MTF assumed*	26%	74%	96%
Probability of computer *success* with 20 million hours MTF assumed*	87%	97%	99.6%

* See text above.

25

32

V CIRCUIT TECHNIQUES

A. GENERAL

In addition to the breadboard circuit discussed above, several other circuits were investigated. These circuits do not constitute complete transfer circuits, but are individual parts which are potentially applicable to a transfer circuit. These circuits were evolved while the breadboard circuit was being developed but for one reason or another were not incorporated in the final circuit. One should not conclude that since they were not incorporated they are necessarily inferior to the techniques used in the final circuit. Rather, they are alternative approaches which might indeed lead to a more desirable transfer circuit than the one illustrated in Fig. 1.5, if further effort were spent in this direction.

B. FIVE-CORE CURRENT STEERING SWITCH

Earlier in this portion of the report the possibility of using two parallel signal channels was discussed, in connection with Fig. 1.3. In such an approach the output of two blocking oscillators is combined to feed into a current steering switch. This switch cannot be the one that forms a part of the original transfer circuit, for there a short in one transistor would disable the entire transfer circuit. An apparently novel means of combining the output of two blocking oscillators is to use the five-core current steering switch shown in Fig. 1.6. An interesting variation of the five-core switch is effected by replacing four of the toroids with two Multi-Aperture-Devices (MAD's). The switch then comprises one toroid and two MAD's. The circuit using the five cores has been successfully operated; no attempt has been made to operate the one composed of MAD's.

In this five-core switch, as in the two-core switch circuit used by JPL, the blocking oscillator pulse overrides the READ clock pulse. This means that the current from the blocking oscillator results in an mmf that is either greater in magnitude or longer in duration than that from the clock pulse, or both. To understand the operation of this switch, first take the case where no blocking oscillator pulse has occurred so that at

33

"FALSE" LOAD   "TRUE" LOAD

SET DIRECTION FOR CORES

CORE E N_1 N_2

CORE D N_1 N_2 N_2 N_1 CORE C

CORE B N_1 N_2 N_2 N_1 CORE A

READ CLOCK   B.O. #1   B.O. #2   INHIBIT CLOCK   B.O. #2   B.O. #1   READ CLOCK
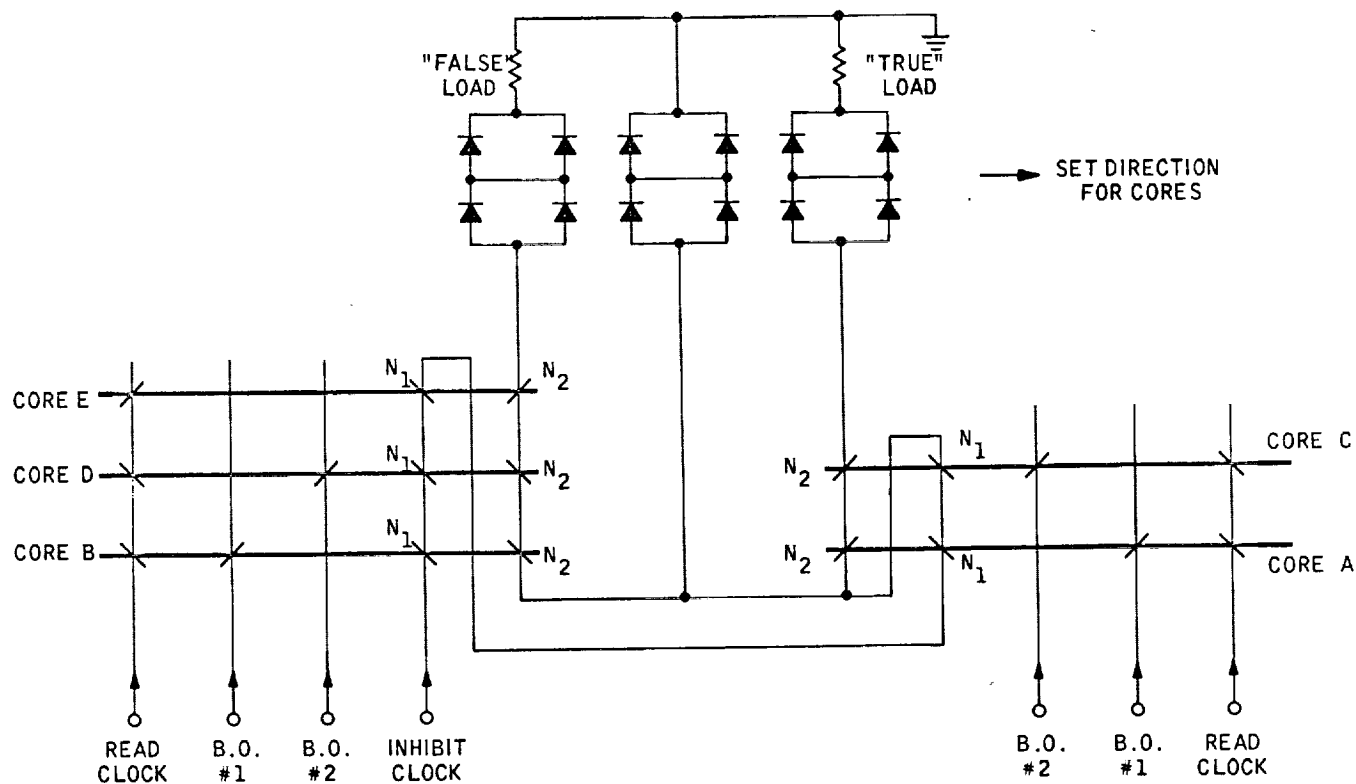
RB-3196-79

FIG. 1.6 FIVE-CORE CURRENT-STEERING SWITCH

28

READ time cores A, B, C, and D are either switched or held in the RESET or CLEAR state, and the E core is switched to the SET state. At a later time, when the INHIBIT pulse is present, Core E is switched to the CLEAR state and current is steered into the "false" branch of the circuit in the conventional manner. Now assume that Blocking Oscillator 1 has an output during READ time; then Cores A, B, and E are all SET at this time. Then the next time an INHIBIT pulse occurs, these three cores will be CLEARED. The voltages induced in the diode loop as a result of this switching are such that the current is steered into the "true" branch. This happens because the voltages from B and D cancel in the diode loop. It is significant to note that the only mmf applied to Cores B and E, in the ideal circuit, is that due to the INHIBIT current going through the windings N_1. This means that the diode loop behaves as if only Core A were switching. Now if both blocking oscillators have an output pulse at READ time, all cores are SET at this time. This time, when the INHIBIT pulse occurs, the voltage from both Core A and Core C try to turn-on the diodes in the "true" output line and the net voltage from Cores B, D, and E is in the direction to maintain the diodes in the "false" line in the nonconducting state. This causes the current to be steered into the "true" output. Note that the voltage in the output loop is treated in two parts—that due to the three cores to the left and that due to the two cores to the right. This is because the cores at the left all have the same mmf applied to them at INHIBIT time, so they all switch at the same rate and therefore have identical voltage pulse shapes and amplitudes. Likewise, the mmf applied to Core A is the same as the mmf applied to Core C, but this value is less than the mmf applied to the three cores at the left.

So far the explanation has considered only normal operation or the absence of a pulse from one blocking oscillator—i.e., a false 0 from one blocking oscillator. What happens when a blocking oscillator has a false 1 by virtue of a short from the emitter to the collector? This is a more difficult fault to deal with unless a fuse is used.* Three possible ways of obtaining failsafe operation without using a fuse were

---

*
The possibility of using fuses was investigated. Microlectron, Inc., Santa Monica, California has developed some Precision Current Limiters under NASA Contract No. NAS 5-2780. These fuses are made in three sizes: 1/16, 1, and 3 ampere ratings. The indications are that these fuses are reliable and could be used to advantage in an application such as this. Further consideration of these fuses seems warranted.

thought of and are indicated in Fig. 1.7. In each of these the direct current through a shorted transistor is diverted into another path so that it doesn't SET the cores in the switch. Except possibly for the circuit shown in Fig. 1.7(c) the operation is apparent by inspection. In Fig. 1.7(c) the steady state value of the direct current can be made to flow almost entirely through the path having the inductor in it. Alternatively, by matching resistance values, the current in this branch can be made to equal the current flowing in the resistive branch so that the net mmf applied to the switch cores is zero. The circuit shown in Fig. 1.7(c) may have some advantages over the simpler one in Fig. 1.7(b), in terms of transient behavior. None of the circuits shown schematically in Fig. 1.7 has been designed or tested. For a specific application it may be possible to reduce the number of diodes in Fig. 1.7(c) below the number shown.

The five-core switch in combination with a dc failsafe circuit should make it possible to successfully combine the output from two channels.

C. AMPLIFIER WITH BASE STROBING

The original non-redundant circuit shown in Fig. 1.1 indicates a strobe is applied to the collector of the amplifier. The purpose of this strobe is to activate the amplifier and hence the transfer circuit only at READ time. The strobe at the collector terminal causes two problems. First of all, if there is a collector-emitter short then a false *1* is generated by the transfer circuit. The second problem is not really a redundancy problem but nevertheless affects the over-all system reliability. The strobe in the collector must of course carry the sum of the collector current from all the amplifiers connected to it. If it were possible to strobe the base rather than the collector, then the current required per amplifier from the strobe circuit would be considerably reduced and the total number of transistors required in the system strobe circuits would be reduced. For these two reasons the possibility of using base-circuit strobing in a redundant circuit was investigated.

There are two features of the base strobing approach that merit special mention. First, since changes in the strobe circuit are being contemplated, attention will be given to its reliability at the same time, even though it is not a part of the transfer circuit. In the circuits to be discussed this is reflected in the use of two transistors

36

(a)

(b)

RB-3196-80

(c)
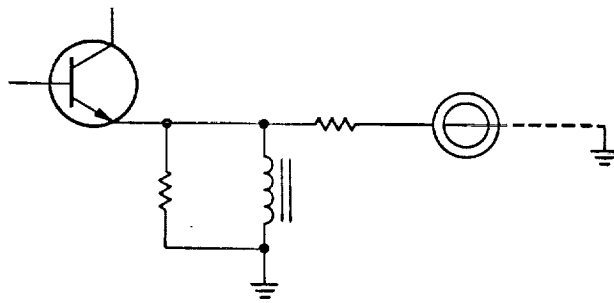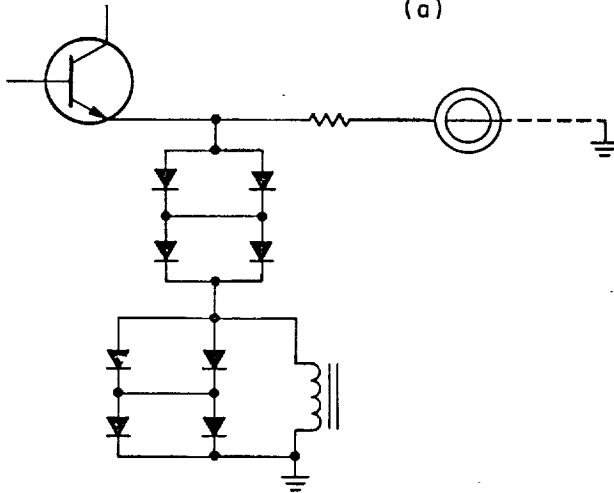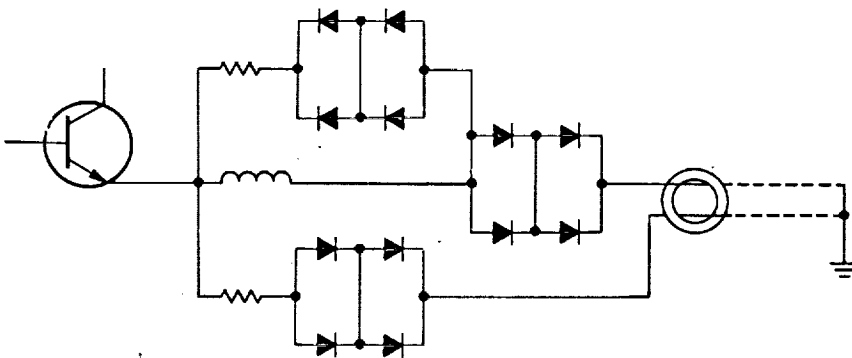
FIG. 1.7   DC FAILSAFE CIRCUITS

37

in series.   If only one transistor were used, a short in it from collector
to emitter would cause an output pulse to occur with each READ pulse.
The second feature is that the voltage which the strobe circuit presents
to the transistor amplifier can no longer vary between wide limits.   The
strobe circuit must apply a voltage that is great enough to allow a *1*
signal to be amplified and at the same time it must be less than the
voltage required to overcome the threshold of the base-emitter junction.
If the strobe voltage is greater than the base-emitter threshold then
the transistor will have an output pulse present whenever the READ clock
pulse appears.   These voltage limits obtain with power supply variations
and ambient temperature variations.

In Fig. 1.8 are shown four of the several circuits that we proposed
and considered.   In the circuit shown in Fig. 1.8(a) the amplifier is
biased off except when the READ clock saturates the strobe transistors.
The saturation of the strobe transistors returns the base of the amplifier
transistor to a negative voltage that is equal to the sum of the two
saturation voltages.   This activates the amplifier.   This circuit fails
to operate satisfactorily for this application because of the presence
of the capacitor in the base circuit.   The capacitance of this capacitor
must be high so that the sense-signal voltage drop across the capacitor
is negligible.   When this condition is met, the RC time constant for
discharging the capacitor is so large that the voltage at the base of
the amplifier does not change significantly from its biased-off value
during the READ pulse time, and therefore the amplifier is not activated.

To eliminate the undesirable effect of the capacitor, the circuit
shown in Fig. 1.8(b) was tried.   Instead of the sense signal and the
strobe voltage being fed into the base in parallel as they are in the
above circuit, they are put in series with each other.   This second
circuit does not operate satisfactorily either, again because of
capacitance.   In this instance it is the internal transistor capacitance
between the base and the emitter that causes the difficulty. This
capacitance is indicated by dashed lines in the figure.   When the strobe
transistors are saturated by the READ clock, the charge that is on the
capacitor starts to discharge.   The discharge path includes the
inductance of the sense signal input transformer, so the capacitance
and inductance form a resonant circuit which rings.   When the voltage
across the capacitor reverses polarity and reaches the threshold value
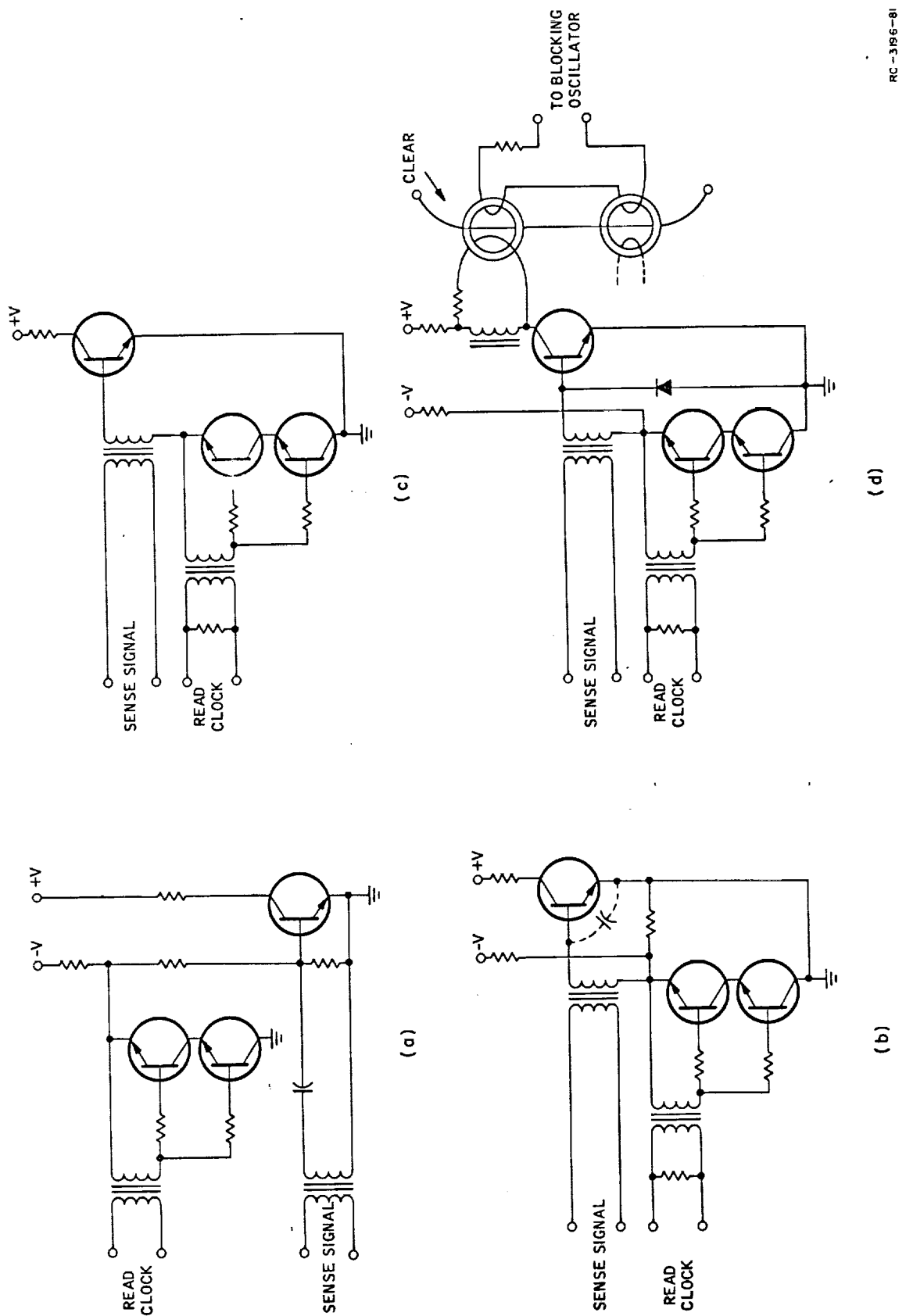
FIG. 1.8 ALTERNATIVE STROBING SCHEMES

(a)

(b)

(c)

(d)

SENSE SIGNAL

READ CLOCK

TO BLOCKING OSCILLATOR

CLEAR

RC-3196-81

33

of the base-emitter junction, the amplifier is put into its active region and an output pulse appears—*i.e.*, a false *1*.

The circuit shown in Fig. 1.8(c) is the same as that in (b), except that the negative bias voltage and bias resistors have been removed. The base-emitter capacitance is no longer charged to the bias value, so there is no ringing when the READ clock pulse appears. However, removing the bias allows some signal to be amplified when the circuit is supposed to be inactive. The signal leaks through the capacitance of the strobe transistors when they are off. Only a small signal is impressed across the amplifier terminals, but nevertheless some output is obtained.

In Fig. 1.8(d) is shown a circuit that brief testing indicates may operate satisfactorily. The bias voltage has been put back in the circuit and a diode added between the base and emitter terminals. The diode prevents the ringing from building up. This circuit does not, however, fulfill one of the initial objectives which led us to consider base strobing—namely, it does not permit a reduction in the number of transistors required in the strobe circuits. It is true that the current which one amplifier circuit requires from the strobe transistor is low (by the factor β) compared to the collector strobing scheme; however, the two series transistors can be used to turn on only one transistor amplifier instead of many. That this is necessarily true can be seen by examining the circuit operation in the event that the amplifier transistor fails by having a short between its base and emitter. The dc potential at the junction of the strobe transistor emitter and the bias resistor is then zero. This failure of one transistor would in effect remove the strobing from all amplifiers that have strobe transistors in common with this faulty amplifier. It would enable these amplifiers to amplify at all times. This circuit has merit but it must be evaluated on the basis that each amplifier would have associated with it its own pair of strobe transistors.

There is one more feature of note in the circuit shown schematically in Fig. 1.8(d). The output of the transistor amplifier is connected to a toroid winding, and there is another toroid associated with this part of the circuit. These toroids form a "combining" circuit that will take the output from two amplifiers and produce one output; from the logic point of view this is an OR circuit. As a general rule it is desirable to combine the signals from two separate channels as often as feasible

34

in order to increase the number of failures that can be tolerated. This assumes, of course, that the combining circuit is highly reliable, as it would be here. The inductor shown in parallel with the toroid is a dc failsafe feature similar to those shown in Fig. 1.7. This "combining" circuit was tested and found to operate satisfactorily.

D. MAGNETIC AMPLIFIER

At one point in the project a cursory examination was given to the possibility of applying a magnetic amplifier to the transfer circuit. In particular, a Ramey[13] type circuit was considered in lieu of the transistor blocking oscillator. It appeared that this circuit might be used advantageously but would require several diodes. Further examination of this circuit seems desirable.

41

VI   SUMMARY AND CONCLUSIONS

This work represents a practical contribution toward implementing a highly reliable computer using Inhibit-Core logic. A reliable transfer circuit was developed and tested. This circuit is reliable because redundancy is used to protect against semiconductor failures. A measure of the increased reliability that has been achieved is indicated by a calculation made, on the basis of certain assumptions, which shows that a computer using the original circuit has a 26% probability of functioning properly while a computer using the redundant circuit has a 96% probability of functioning properly. In another instance the two figures are 87% and 99.6% respectively.

Several circuit techniques applicable to achieving high reliability were investigated in addition to developing the breadboard circuit. Some of these techniques rely upon the inherent reliability of ferrite toroids.

This work is incomplete in the sense that some initial assumptions were idealistic, and in the sense that some promising techniques have not been investigated in sufficient depth to fully assess them. The initial assumptions include the supposition that the clock circuit and the strobe voltage circuit are perfectly reliable. It seems appropriate now to investigate the possibility of making these and other circuits reliable through semiconductor redundancy and magnetic circuitry. It also seems desirable to investigate more fully techniques other than those made use of in the breadboard circuit, and to further test and optimize the breadboard circuit.

42

PART TWO: RELIABLE MEMORY TECHNIQUES

By

R. C. Minnick

43

I INTRODUCTION

Considering the coincident-current memory system shown as Fig. 2.1, it is seen that the words are stored as vertical lines, and a particular word is read by selecting a vertical plane along X and another along Y. Writing is accomplished by selecting the same pair of planes with the opposite current, and simultaneously inhibiting or not inhibiting each of the Z planes.

If the goal is to render this memory reliable, there are in general two recognized ways to proceed: by means of component redundancy or by code redundancy. That is, one can either add equipment to the system so that some fraction of the equipment can fail without impairing the overall operation of the memory, or one can let the words stored in the memory, perhaps as well as the addresses, be represented in an error-correcting code. The addition of code redundancy of course implies extra equipment, but for memory systems the use of it usually involves less added equipment than the approach of circuit redundancy.

Thus the input word to the memory of Fig. 2.1 might be represented in some error-correcting code. The inhibit drivers would connect through a combinational code-correction circuit that would do the error correcting. Therefore, it is possible to protect the memory against failures in the inhibit drivers if the words are stored in the form of some error-correcting code. In a very similar way one can protect against failures of the sense amplifiers by installing another identical code-correction circuit that would do the error correcting. This also protects against the unlikely failures in the memory cores themselves. It may be, of course, that only one of these two code-correction circuits is needed. Such code-correction circuits will be considered in Part Two, Sec. IV.

Each access circuit in Fig. 2.1 is assumed to consist of a semiconductor address register and semiconductor drivers, with single-aperture magnetic circuits between. It is further assumed that the problem is to protect against failures in the semiconductors.

It is sufficient to consider just one of the two access switches of Fig. 2.1, as they are similar. Indeed, a word-organized memory would
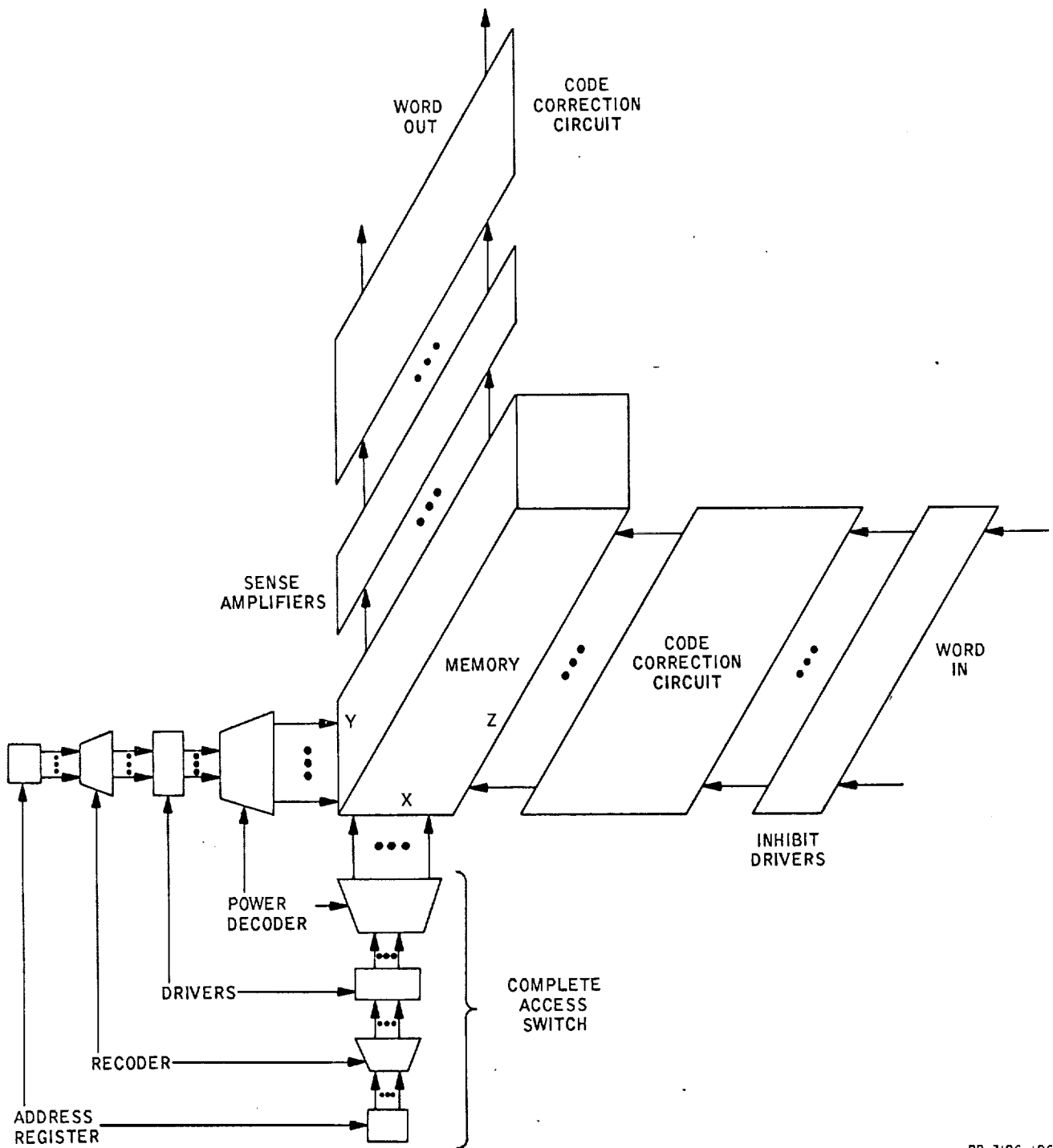
FIG. 2.1 MEMORY SYSTEM

RB-3196-106

45

have only one; and the results apply there. This access switch is shown
in more detail as Fig. 2.2. The switch consists of a *recoder*, which serves
to change the code in the address register into one that is suitable for
operating the power decoder, a set of *transistor drivers*, and the *power
decoder*, which is the usual magnetic access switch.

Techniques for the design of power decoders that allow the access switches
to operate properly, even though no more than some maximum number of transistor
drivers have failed, were developed in the previous year's research on this topic.[3]
In Part Two, Secs. II and III, the design of the recoder will be developed.
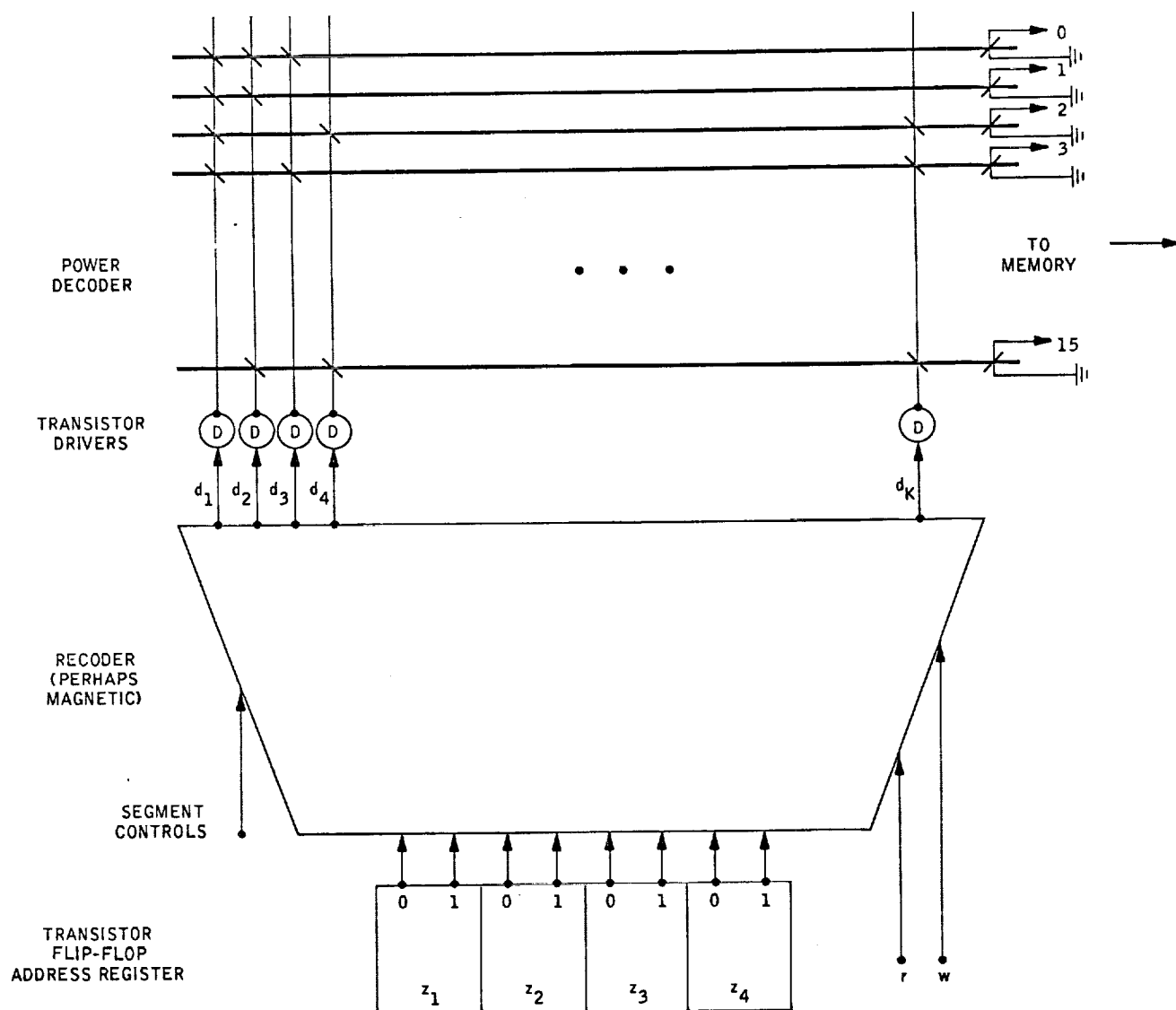
FIG. 2.2   ACCESS SWITCH

43

RB-3196-116

II RECODERS FOR READ ACCESS SWITCHES

A. DEFINITIONS

It is a well-known fact, although it is often ignored, that the recoding necessary to translate the code in the memory address register to the code required at the input of a load-sharing read access switch may be exceedingly complicated. If it is intended to use redundancy external to the memory as well as in the memory, and if the memory designer is free to choose the nature of this redundancy, it has been shown[3] that the recoder can be greatly simplified, or perhaps even eliminated. Unfortunately, however, this assumption is probably unrealistic in some cases. For instance, it may be that no redundancy is allowed external to the memory, but that it is needed within the memory. Or perhaps only one or two bits of redundancy are used in the arithmetic sections, but due to the importance attached to the memory, significantly more redundancy may be needed there.

Referring again to Fig. 2.2, it is assumed that the address in the memory address register is densely coded. Therefore, if the memory is of the coincident-current variety (two-dimensional read, three-dimensional write) it usually is required that there be an even number of bits so that half may refer to each dimension. Furthermore, within each dimension of the memory there may be several access switches arranged so that the outputs of each access switch drive a *segment* of the coordinate memory lines in that dimension. This segmentation of the access switch into several smaller ones enables the designer to keep the number of wires per core in the access switch to a reasonable value, allows him to control the load-sharing factor, and keeps to a practically low value the number of unit magnetomotive forces of one polarity that must be exactly balanced against those of the other polarity.

In addition to receiving signals from the memory address register and the signal controls, the recoder receives a signal r to initiate the *read* operation and w to *write*. The write operation serves to deliver the same but opposite polarity current on the same memory line as did the read operation.
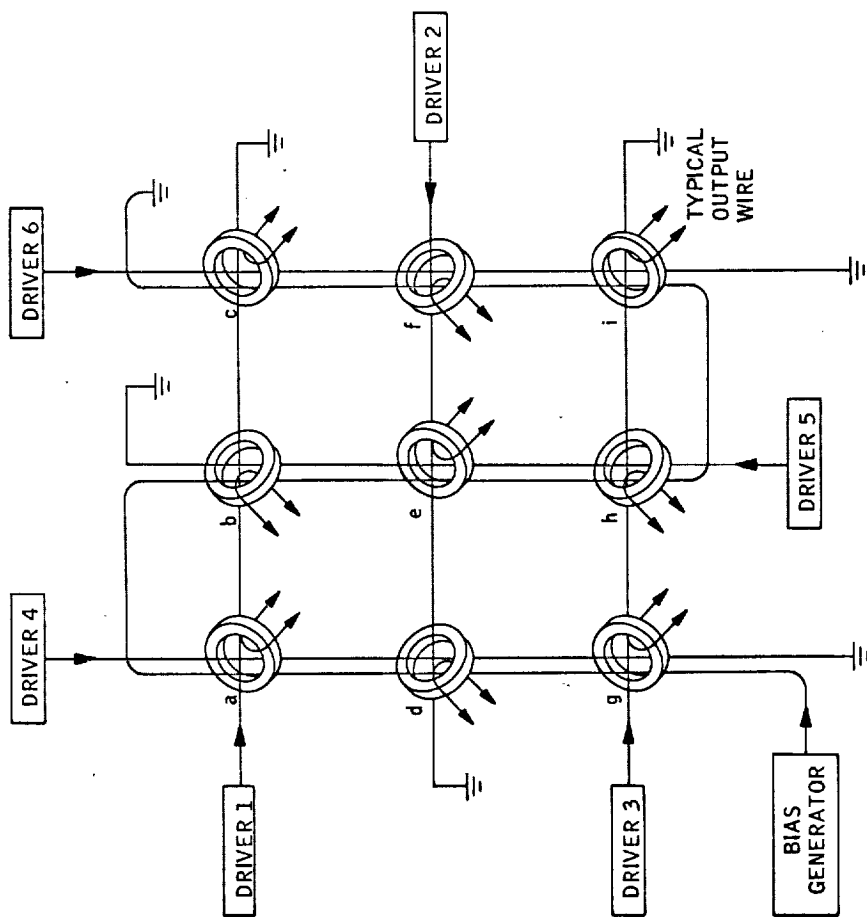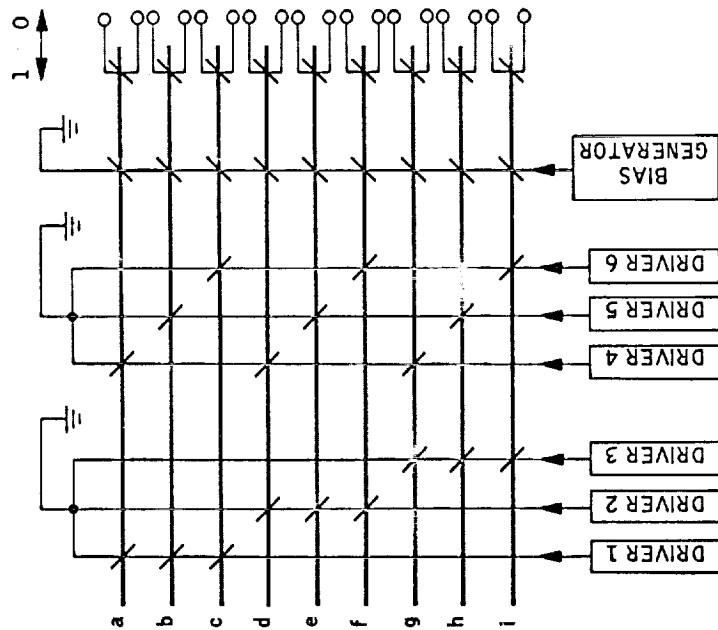
FIG. 2.3   A READ ACCESS SWITCH

RB-3196-115

46

In order to describe the operation of the recoder, it is necessary first to review the description of the power decoder. To begin by means of an example, the nine-output power decoder in Fig. 2.3(a) is called a *two-dimensional switch*, because there are two sets of wires arranged in such a way that exactly one wire in each set is energized. As the geometric picture of a switch becomes difficult when the number of dimensions is increased, it is better to represent a switch in a *mirror* notation where the cores are horizontal lines, the wires are vertical lines, and the slashes indicate whether the wires intersect the cores, and with what polarity. The switch of Fig. 2.3(a) is drawn in this notation as Fig. 2.3(b). It is possible to simplify this notation even more by noting that the gridwork of cores and wires is not needed; only the pattern of slashes is required. Thus one arrives at the $\underline{W}$ or *winding* matrix to represent the wiring pattern of the switch. For the present example, the $\underline{W}$ matrix is

$$\underline{W} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{2.1}$$

It should be noted that for the purpose of simplifying this example, the bias windings have been omitted.

In order to deduce the properties of this switch it is necessary to know its behavior for all acceptable ways of energizing the drivers at the time of the r signal. For this a *read selection* or $\underline{S}$ matrix is defined. Each row represents one driver while each column represents an input code. For the present example the $\underline{S}$ matrix is

$$\underline{S} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{2.2}$$

Thus for the leftmost code, Drivers 1 and 4 of Fig. 2.3 are turned on. If the matrix product $\underline{X} = \underline{W}\underline{S}$ is formed, the upper left term of X is the product of the current delivered by the first driver times the number of turns that drive line makes with the first core plus the same for all the remaining drivers and the first core. For the example of Fig. 2.3, the $\underline{X}$ matrix is

$$
\underline{X} = \underline{W}\underline{S} =
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
$$

$$
=
\begin{bmatrix}
2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2
\end{bmatrix}
\qquad (2.3)
$$

Thus the x_{11} term is the mmf on Core a due to Code State 1. In general, $\underline{X}$ is the *excitation* or mmf matrix which in each column shows the mmfs applied to successive cores for a given code state. By supplying an appropriate bias, all mmfs can be altered by the same amount, so it is apparent for this example that a bias of -1 will assure the selection of a different core for each code state.

A *write selection* matrix, $\underline{S}^*$, may be defined in a similar manner to $\underline{S}$. It represents all acceptable ways of energizing the drivers at the time of the w signal.

The winding matrix $\underline{W}$ may be simplified further. For the example of Fig. 2.3 it is noted that exactly one driver was energized in each

48

of two sets of three drivers. If the $\underline{W}$ matrix of Eq. (2.1) is partitioned into two sets of three columns each, then there is exactly one 1 in each section of each row of $\underline{W}$. The $\underline{W}$ may be condensed into an $\underline{R}$ or *code matrix* by placing a digit corresponding to the position of the 1 in $\underline{W}$. For the present example, $\underline{R}$ turns out to be simply the matrix of the ordered two-place ternary numbers 00 to 22:

$$
\underline{W} = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\quad , \quad
\underline{R} = \begin{bmatrix}
0 & 0 \\
0 & 1 \\
0 & 2 \\
1 & 0 \\
1 & 1 \\
1 & 2 \\
2 & 0 \\
2 & 1 \\
2 & 2
\end{bmatrix}
\quad .
\tag{2.4}
$$

It has previously been shown[3] that the parameters of the read access switch are closely related to the parameters of the redundant code that is applied to the drivers. This relationship is given as Table 2.1.

Table 2.1

RELATIONSHIPS OF CODES TO SWITCHES

CODE	SWITCH
Radix	Number of wires per input set
Number of positions in code word	Number of sets of input wires
Number of nonredundant positions in a code word	Number of nonredundant sets of input wires (the dimension)
Hamming distance	Load-sharing factor

B. RECODERS FOR NON-ZERO-NOISE SWITCHES

In Sec. II-A above, the switch was represented first by the matrices $\underline{W}$, $\underline{S}$, and $\underline{S}^*$; and then these were condensed into the matrix $\underline{R}$. A convenient approach to the design of a switch and the corresponding recoder is to proceed in the opposite order. To illustrate this, the design of a sixteen-output switch is developed below, using the notation of Fig. 2.2.

51

The first step is to choose as the $\underline{R}$ matrix the sixteen four-bit binary numbers, with a single even parity-check bit:

$$
\underline{R} = \left[\begin{array}{cccc|c}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0
\end{array}\right] \quad . \tag{2.5}
$$

By the previous definitions, this $\underline{R}$ matrix leads to the following $\underline{W}$ matrix:

$$
\underline{W} = \left[\begin{array}{cc|cc|cc|cc|cc}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}\right] \quad . \tag{2.6}
$$

50

If the read selection matrix is chosen to be $\underline{S} = \underline{W}^T$, the transpose of $\underline{W}$, then the excitation matrix is found to be

$$\underline{X} = \begin{bmatrix} 5 & & & 1,3 \\ & \cdot & \cdot & \\ & & \cdot & \\ 1,3 & & & 5 \end{bmatrix} . \tag{2.7}$$

The symbolism of Eq. (2.7) indicates that $\underline{X}$ has 5 as each main diagonal element, and either 1 or 3 as each off-diagonal element. It is evident from Eq. (2.7) that a bias of -3 units is required for the read selection. This bias may be included by appending three bias columns of minus ones to Eq. (2.6) and correspondingly by appending three rows of ones to $\underline{S}$. The write selection matrix for this example merely provides for energizing two of the three bias windings alone. With these augmentations, the winding matrix becomes

$$\underline{W}_1 = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & -1 & -1 & -1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & -1 & -1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & -1 & -1 & -1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & -1 & -1 & -1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & -1 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & -1 & -1 & -1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 & -1 & -1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & -1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & -1 & -1 & -1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & -1 & -1 & -1 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & -1 & -1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & -1 & -1 & -1
\end{bmatrix} , \tag{2.8}$$

the read-selection matrix becomes

$$\underline{S}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \end{matrix} \quad , \qquad (2.9)$$

$$
\begin{array}{ll}
z_1: & 0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1 \\
z_2: & 0\;0\;1\;1\;0\;0\;1\;1\;0\;0\;1\;1\;0\;0\;1\;1 \\
z_3: & 0\;0\;0\;0\;1\;1\;1\;1\;0\;0\;0\;0\;1\;1\;1\;1 \\
z_4: & 0\;0\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;1\;1\;1
\end{array}
$$

and the write-selection matrix becomes

$$\underline{S}^*_1 = \begin{bmatrix} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \end{matrix} \quad . \qquad (2.10)$$

$$
\begin{array}{ll}
z_1: & 0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1\;0\;1 \\
z_2: & 0\;0\;1\;1\;0\;0\;1\;1\;0\;0\;1\;1\;0\;0\;1\;1 \\
z_3: & 0\;0\;0\;0\;1\;1\;1\;1\;0\;0\;0\;0\;1\;1\;1\;1 \\
z_4: & 0\;0\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;1\;1\;1
\end{array}
$$

In order to establish the recoder equations, it is convenient to list the recoder outputs, d_i, along the right side of $\underline{S}_1$ and $\underline{S}_1^*$, and the recoder inputs, z_i, along the bottom. Then from Eqs. (2.9) and (2.10), the recoder equations for this example are found to be

$$d_1 = z_1'r \, , \qquad\qquad d_2 = z_1 r \, ,$$

$$d_3 = z_2'r \, , \qquad\qquad d_4 = z_2 r \, ,$$

$$d_5 = z_3'r \, , \qquad\qquad d_6 = z_3 r \, ,$$

$$d_7 = z_4'r \qquad\qquad d_8 = z_4 r$$

$$d_9 = (z_1 \oplus z_2 \oplus z_3 \oplus z_4)'r \, , \qquad d_{10} = (z_1 \oplus z_2 \oplus z_3 \oplus z_4)r \, ,$$

$$d_{11} = d_{12} = r + w \, , \qquad\qquad d_{13} = r \quad . \qquad\qquad (2.11)$$

This switch is seen to have a load-sharing factor of two, and thirteen drivers are required. Any one of Drivers d_1, d_2, ... d_{10} can fail by open circuiting, and the switch will still operate correctly, but with reduced output power. This switch will not operate correctly upon failure of Drivers d_{11}, d_{12} or d_{13}.

An obvious way to obtain more redundancy and to provide more protection against driver failures is to increase the distance of the code. For instance, one might consider the (7,3) Hamming code. As this is a seven-bit code with a distance of three, it is possible to deduce immediately that the most positive off-diagonal element of $\underline{X}$ before biasing the switch, is four. Thus, minus four units of bias are required for the read selection operation, while minus three units are needed for the write selection. Following the same procedures that were used for the previous example, the augmented winding, read-selection, and write-selection matrices may be deduced from the code matrix, and the recoder equations may be found. Omitting the intervening steps, the $\underline{R}$ matrix and the recoder equations for the sixteen-output switch based on the (7,3) Hamming code are found to be:

$$\underline{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (2.12)$$

and

$$d_1 = z_1' r , \qquad\qquad d_2 = z_1 r ,$$

$$d_3 = z_2' r , \qquad\qquad d_4 = z_2 r ,$$

$$d_5 = z_3' r , \qquad\qquad d_6 = z_3 r ,$$

$$d_7 = z_4' r , \qquad\qquad d_8 = z_4 r ,$$

$$d_9 = (z_1 \oplus z_2 \oplus z_4)' r , \qquad d_{10} = (z_1 \oplus z_2 \oplus z_4) r ,$$

$$d_{11} = (z_1 \oplus z_3 \oplus z_4)' r , \qquad d_{12} = (z_1 \oplus z_3 \oplus z_4) r ,$$

$$d_{13} = (z_2 \oplus z_3 \oplus z_4)' r , \qquad d_{14} = (z_2 \oplus z_3 \oplus z_4) r ,$$

$$d_{15} = d_{16} = d_{17} = r + w , \quad d_{18} = r . \qquad (2.13)$$

This (7,3) Hamming switch has a load-sharing factor of three, and it uses eighteen drivers. It is insensitive to any open-circuit failure or to any short-circuit failure of any driver. For instance, if one of the drivers, d_1, d_2, ..., d_{14}, is open-circuited, the select mmf is reduced from three to two. If one of the bias drivers, d_{15}, ..., d_{18}, is open-circuited, a non-selected core that should get zero units now gets one unit of mmf, but it still does not switch. In a similar manner short-circuit failures of the drivers are protected.

54

C. CONSTANTINE SWITCH RECODERS

A Constantine switch or *C switch* is an obvious choice for a reliable access switch that operates from a dense address code because such a switch has a power of two outputs, has zero noise as well, and is load-sharing. The winding matrix, $\underline{W}_n$, and the read selection matrix, $\underline{S}_n$, for the *C* switch having 2^n outputs may be written as[†]

$$
\underline{W}_n = \begin{bmatrix} \underline{W}_{n-1} & \underline{W}_{n-1} \\ \underline{W}_{n-1} & -\underline{W}_{n-1} \end{bmatrix} , \qquad
\underline{S}_n = \begin{bmatrix} \underline{S}_{n-1} & \underline{S}_{n-1} \\ \underline{S}_{n-1} & \underline{V}_{n-1} - \underline{S}_{n-1} \end{bmatrix} ,
$$

where

$$
\underline{W}_1 = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix} , \qquad
\underline{S}_1 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} , \qquad (2.14)
$$

and

$$
\underline{V}_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & 1 \\ & & \cdots & \\ 1 & & & 1 \end{bmatrix} \Big\} \, 2^{n+1} \quad .
$$

$$
\longleftarrow 2^n \longrightarrow
$$

The excitation matrix, $\underline{X}_n$ is found by multiplying $\underline{W}_n$ and $\underline{S}_n$:

$$
\underline{X}_n = \underline{W}_n \underline{S}_n = \begin{bmatrix} 2\underline{W}_{n-1}\underline{S}_{n-1} & \underline{W}_{n-1}\underline{V}_{n-1} \\ & \\ \underline{0}_{n-1} & 2\underline{W}_{n-1}\underline{S}_{n-1} - \underline{W}_{n-1}\underline{V}_{n-1} \end{bmatrix} . \qquad (2.15)
$$

[†] The notation used here is briefly described in Ref. 3, and in more detail elsewhere by Minnick and Haynes.[14]

57

where $\underline{0}_{n-1}$ is the $2^{n-1} \times 2^{n-1}$ zero matrix. The product $\underline{W}_{n-1}\underline{V}_{n-1}$ in Eq. (2.15) has elements that are sums of the rows of $\underline{W}_{n-1}$, because $\underline{V}_n$ is the matrix of all 1's. But from Eq. (2.14) the sums of both rows of $\underline{W}_1$ are zero, and from the method of formation of $\underline{W}_n$ in terms of $\underline{W}_1$ the sums of the rows of $\underline{W}_n$ also are zero. Hence $\underline{W}_{n-1}\underline{V}_{n-1} = \underline{0}_{n-1}$. This reduces Eq. (2.15) to

$$\underline{X}_n = \begin{bmatrix} 2\underline{X}_{n-1} & \underline{0}_{n-1} \\ \\ \underline{0}_{n-1} & 2\underline{X}_{n-1} \end{bmatrix} \quad . \qquad (2.16)$$

Equation (2.16) may be written as

$$\underline{X}_n = 2^n \underline{I}_n \qquad (2.17)$$

where I_n is the $2^n \times 2^n$ unit matrix.

To write in the memory, it is necessary to generate an equal magnetomotive force of the opposite polarity at output of the selected core of the C switch. This may be done by energizing the C-switch drivers in accordance with a write-selection matrix $\underline{S}_n^*$ which is defined as

$$\underline{S}_n^* = \underline{V}_n - \underline{S}_n \quad . \qquad (2.18)$$

The write-excitation matrix is

$$\underline{X}_n^* = \underline{W}_n \underline{S}_n^* = \underline{W}_n \underline{V}_n - \underline{W}_n \underline{S}_n$$

$$= -\underline{X}_n \qquad (2.19)$$

as required. An examination of Eq. (2.17) shows that a C switch that is to decode n address register bits to one of 2^n outputs has a load-sharing factor of 2^n, and that it is indeed zero noise. The nature of the recoder required for this switch is indicated by the structure of the read-selection matrix, $\underline{S}_n$, and the write-selection matrix, $\underline{S}_n^*$, in Eq. (2.18). Supposing the access switch drivers (or correspondingly the outputs of the recoder) are named d_1, d_2, $\ldots$, d_{2^n+1}; then, as in Sec. II-B, each

of the 2^{n+1} rows of $\underline{S}_n$ may be labelled by a driver and each of the 2^n columns of $\underline{S}_n$ may be labelled by a code combination in the address register. In particular, letting z_1, z_2, ..., z_n be the address register bits, $\underline{S}_1$ and $\underline{S}_2$ may be labelled as follows:

$$\underline{S}_1 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} \quad , \qquad \underline{S}_2 = \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array} \right] \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{matrix} \quad . \qquad (2.20)$$

$$z_1: \quad 0 \quad 1$$

$$z_1: \quad 0 \quad 1 \quad 0 \quad 1$$
$$z_2: \quad 0 \quad 0 \quad 1 \quad 1$$

By inspection of Eq. (2.20), the recoder equations for $\underline{S}_1$ are seen to be

$$d_1 = 0 \, , \quad d_2 = 1 \, , \quad d_3 = z_1 \, , \quad d_4 = z_1' \quad . \quad (2.21)$$

For the $\underline{S}_n$ recoder it is observed that from the method of formation of $\underline{S}_n$ from $\underline{S}_{n-1}$ [Eq. (2.14)] and from the order that was chosen for the address register code combinations, the variable z_n is redundant in the first half of the equations for d_i. Similarly, each equation for d_i in the second half of $\underline{S}_n$ is the exclusive-OR of the corresponding equation in the first half and of the new variable, z_n. That is,

$$d_i = \begin{cases} d_{i-4} \oplus z_2 \, , & i = 5, 6, \ldots, 8 \\[2mm] d_{i-8} \oplus z_3 \, , & i = 9, 10, \ldots, 16 \\[2mm] \ldots \\[2mm] d_{i-2^n} \oplus z_n \, , & i = 2^n + 1, \, 2^n + 2, \, \ldots, \, 2^{n+1} \, . \end{cases} \quad (2.22)$$

Equations (2.21) and (2.22) give the C-switch recoder equations for generating the memory *read* pulse. From Eq. (2.18) the write selection matrix is the complement of the read selection matrix; therefore, the

recoder equations for the write operation are the complements of Eqs. (2.21) and (2.22). Introducing the symbols r and w for the read and write commands, and noting that $\dot{w} = r'$, Eq. (2.21) generalizes to

$$d_1 = r', \quad d_2 = r, \quad d_3 = (r \oplus z_1)', \quad d_4 = r \oplus z_1, \qquad (2.23)$$

and Eq. (2.22) remains unchanged.

Equations (2.22) and (2.23) define the C-switch recoder for the read and write operations. Explicitly, the values of d_i for $n = 2$ are

$$
\begin{aligned}
d_1 &= r', & d_2 &= r, \\
d_3 &= (r \oplus z_1)', & d_4 &= r \oplus z_1, \\
d_5 &= (r \oplus z_2)', & d_6 &= r \oplus z_2, \\
d_7 &= (r \oplus z_1 \oplus z_2)', & d_8 &= r \oplus z_1 \oplus z_2.
\end{aligned}
\qquad (2.24)
$$

It is evident from Eqs. (2.22), (2.23), and (2.24) that the C-switch recoder must produce all possible exclusive-ORs and their complements that have r as one input, and each possible combination of the address register bits as the other inputs. As the logical design of the networks, which will be termed *linear trees*, to produce these exclusive ORs seems to be of practical importance, they will be developed now.

Considering a diode-type two-variable exclusive-OR gate, each of the two variables as well as their complements must be provided. Since, in linear trees certain gate outputs must serve as inputs to other gates, it is necessary that each diode-type gate produce the exclusive-OR as well as its complement. Therefore, it is assumed that such *complementary* two-variable exclusive-OR gates are available; while they will be symbolized as having two input leads and one output lead, it is understood that in fact each gate has four input leads and two output leads. It also follows that only the d_i for even values of i need to be produced.

There are 2^n outputs of an n variable linear tree. One of these, $d_2 = r$, is trivial; hence, the minimum number of gates needed in the linear tree is $2^n - 1$. A tree using this minimum may easily be defined as follows: The first (or input) level of the tree has n gates, each having r as one input and a different one of the address register variables as the other input. The second level has C_2^n gates arranged in

such a way that one input comes from one of the first-level outputs and
the other comes from one of the address register variables. The inputs
to the second level are chosen in such a manner that the second-level
outputs are all three-input exclusive-ORs of r and any two of the vari-
ables. The successive levels of the tree are arranged in a similar
manner. Level j of the tree has C_j^n gates, for $j = 1, 2, \ldots, n$, and there
are n levels in all. The number of gates is $\sum_{j=1}^{n} C_j^n = 2^n - 1$, which is
the minimum previously specified. Trees of this type will be called
minimum-gate linear trees. Minimum-gate linear trees for $n = 2, 3, 4,$
and 5 are drawn as Figs. 2.4 and 2.5. Each variable z_i is indicated
on these and following figures by i, and the exclusive-ORs $z_i \oplus z_j \oplus$
$\ldots \oplus z_k \oplus r$ are simplified to $ij \ldots kr$.

While the minimum-gate linear trees that have been described con-
tain the minimum number of gates, they also have more than the minimum
number of levels. If p is chosen as the least integer such that
$n \leq 2^p - 1$, then a different linear tree having p levels can be built
of two-variable exclusive-OR gates; however, such a tree will in general
have more than $2^n - 1$ gates. A tree of this type is called a *minimum-
level linear tree*. In many practical situations the saving in the
number of cascaded levels can more than justify the additional gates.

In order to construct minimum-level linear trees, it is convenient
first to solve a simpler problem. This is, given n variables, to con-
struct a minimum-level network that provides all possible linear sums
of the variables. For convenience, this network is termed an *all-function
network*. In this case it is not necessary to provide the r signal that
is common to all outputs, and this simplification makes all the differ-
ence. Again there are 2^n network outputs, but this time any network
output can be constructed from two other network outputs of fewer
variables. Furthermore, for $2^{p-1} < n < 2^p$, the two components forming
any network output need not contain more than p variables each; or, if
$n = 2^{p-1}$, the two components need not contain more than $p - 1$ variables
each. Hence, the network can be constructed with only one element per
output within the minimum number of levels. Also, since the null and
single-variable outputs can be obtained directly, the total number of
elements required for the all-function network is just $2^n - n - 1$.

Using this all-function network it is possible to partition the
original problem, which does require the common variable r for each
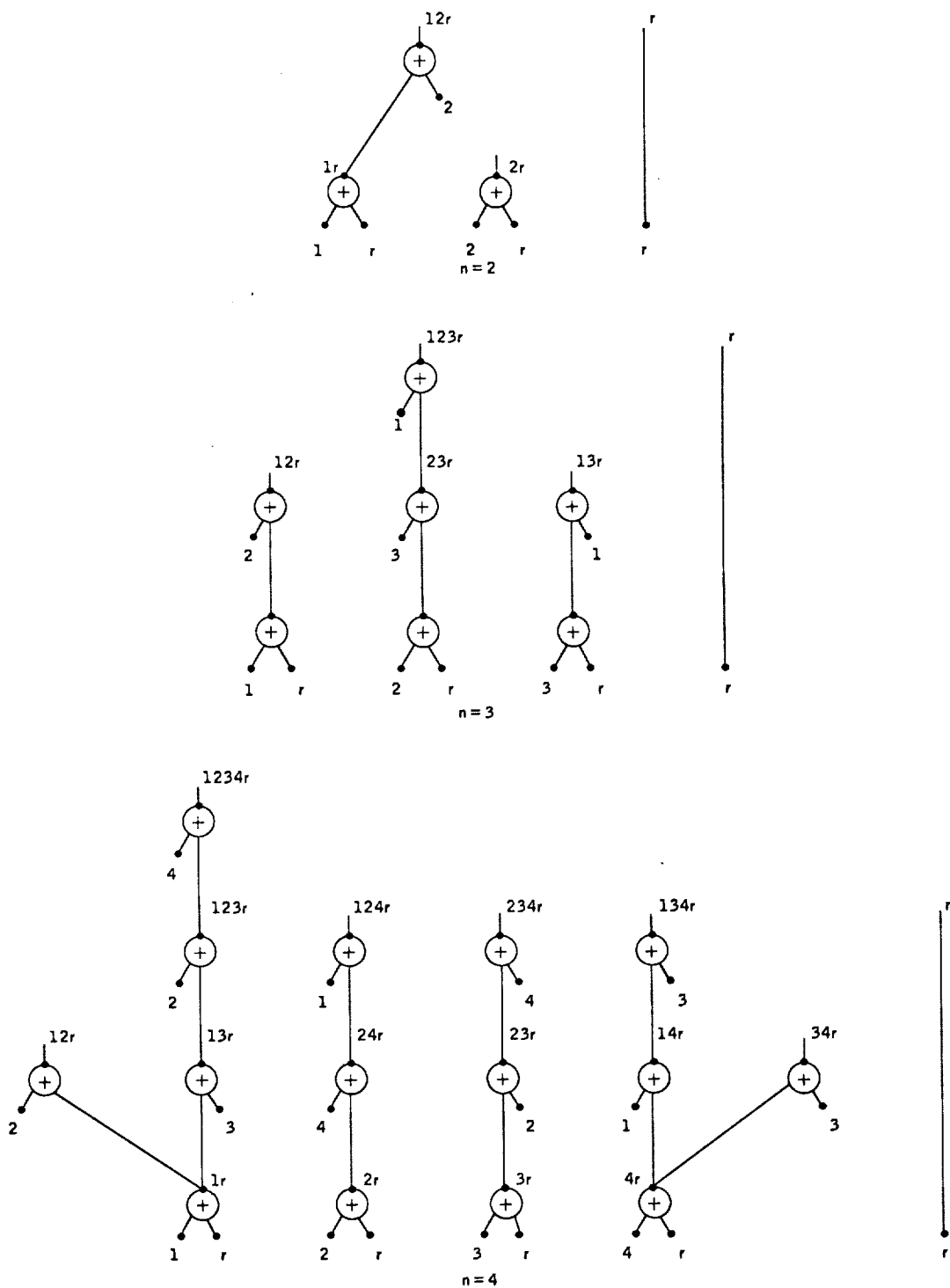
FIG. 2.4   MINIMUM-GATE LINEAR TREES, n = 2, 3, 4

RB-3196-108

62
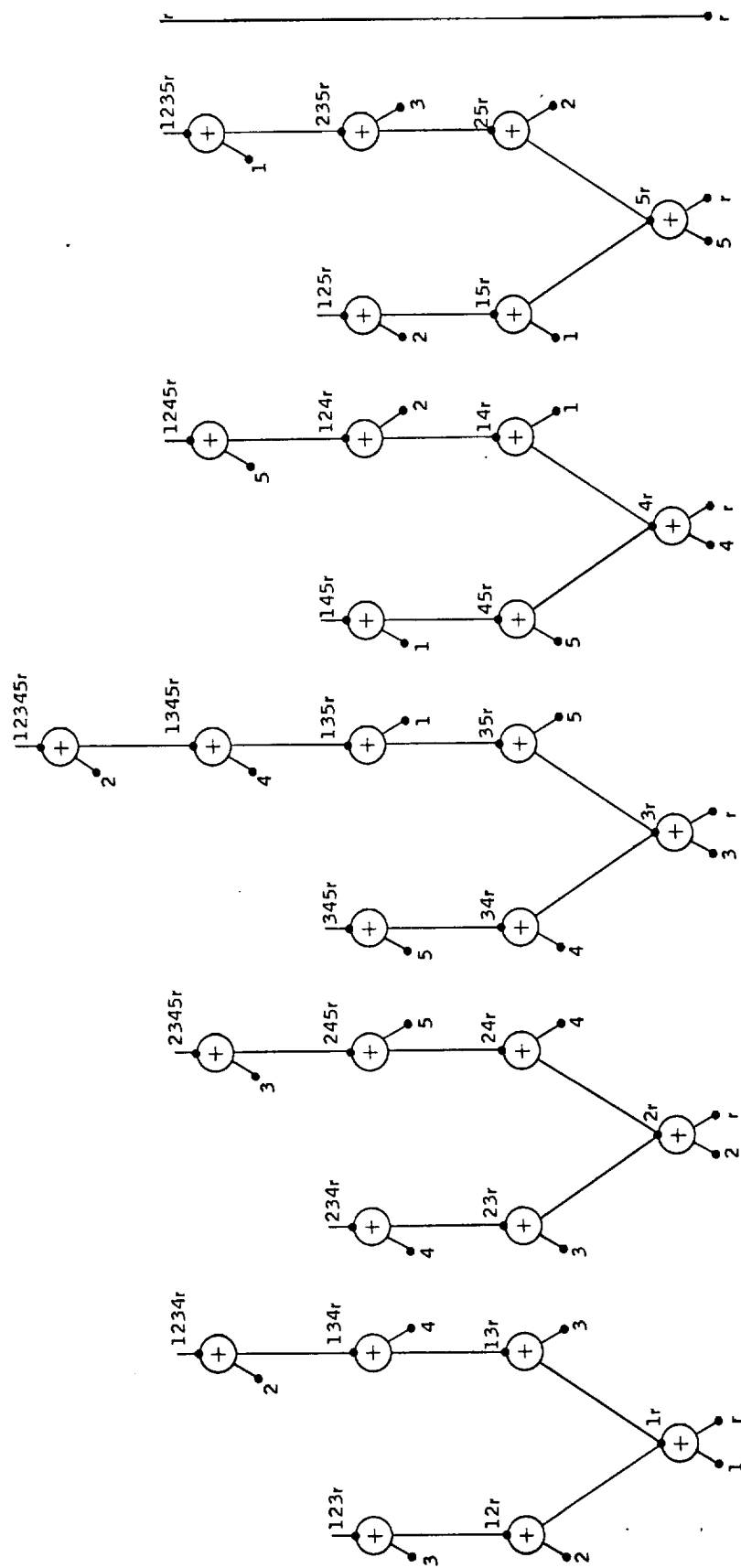
FIG. 2.5   MINIMUM-GATE LINEAR TREE, n = 5

RB-3196-105

63

network output. Supposing first that $n = 2^p - 1$, i.e., the total number of variables is a power of two, the variables are partitioned into two equal subsets. For the first subset, which contains the variable r, a minimum-level decoding tree in $2^{p-1} - 1$ variables is specified. That is, the original problem with p indexed downward once. For the second subset an all-function network in 2^{p-1} variables is constructed. Both subnetworks use no more than $p - 1$ levels. Every network output not already realized can be composed within exactly one more level (and one gate each) from the outputs of these two subnetworks. The situation is summarized in Fig. 2.6. Since the only non-output gates occur in the all-function network, one can enumerate them and reduce the order of the problem. Proceeding iteratively, the realization of the $2^{p-1} - 1$ variable subnetwork is treated next. By enumerating the number of non-outputs required at each step, it is possible to obtain an upper bound on the number of elements required.
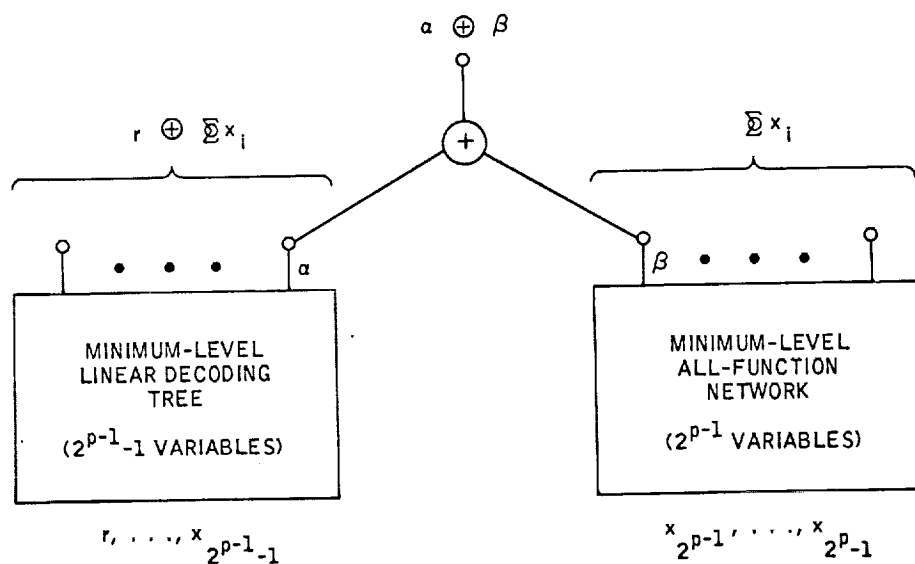
FIG. 2.6   PARTITION OF NETWORK FOR $n = 2^p - 1$

For other values of n such that $n = m + 2^{p-1} - 1$, one can prescribe that the first partition is to isolate only m variables in the all-function network, and then it is possible to proceed as before. As a result, for any $n = m + 2^{p-1} - 1$, a minimal-level linear decoding tree can be systematically developed with

62

$$N_n \leq 2^n - 1 + \sum_{k=1}^{p-2} \left(2^{2^{p-k-1}} - 2^{p-k-1} - 1 \right) + 2^m - m - 1$$

or

$$N_n \leq 2^n + 2^m - m - 2 + \sum_{k=1}^{p-2} \left(2^{2^{p-k-1}} - 2^{p-k-1} - 1 \right) \quad . \qquad (2.25)$$

Examples of networks so constructed are shown as Figs. 2.7 and 2.8. In Table 2.2 pertinent evaluations for low values of n are listed.

For values of $n \leq 7$ it can be shown that the bounds in Table 2.2 are actually minimum values. Furthermore, the entry for $n = 8$ is minimal since it requires no more non-output elements than for $n = 7$. Beyond this, neither proofs nor counterexamples have been constructed, so nothing else about minimality is asserted.
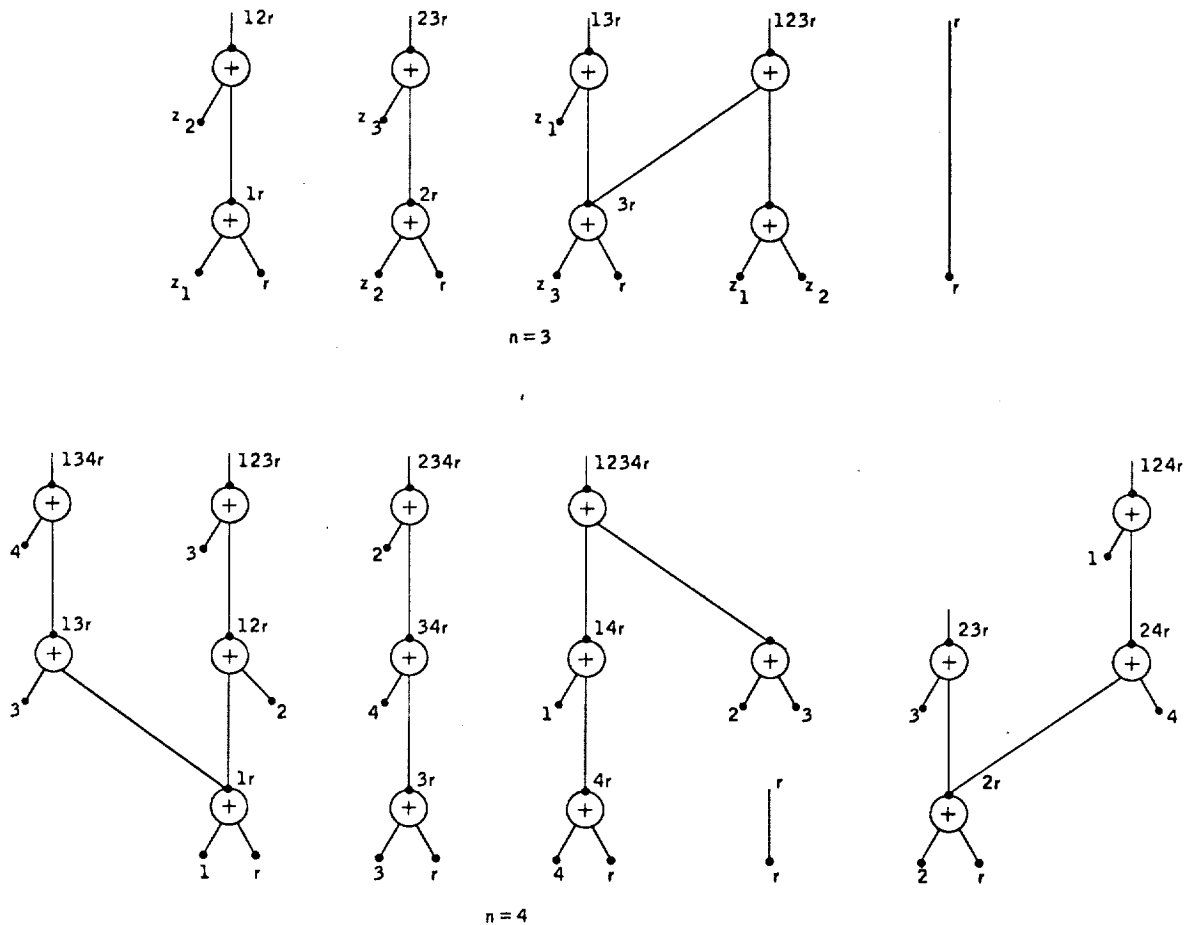
FIG. 2.7 MINIMUM-LEVEL LINEAR TREES, n = 3, 4

63
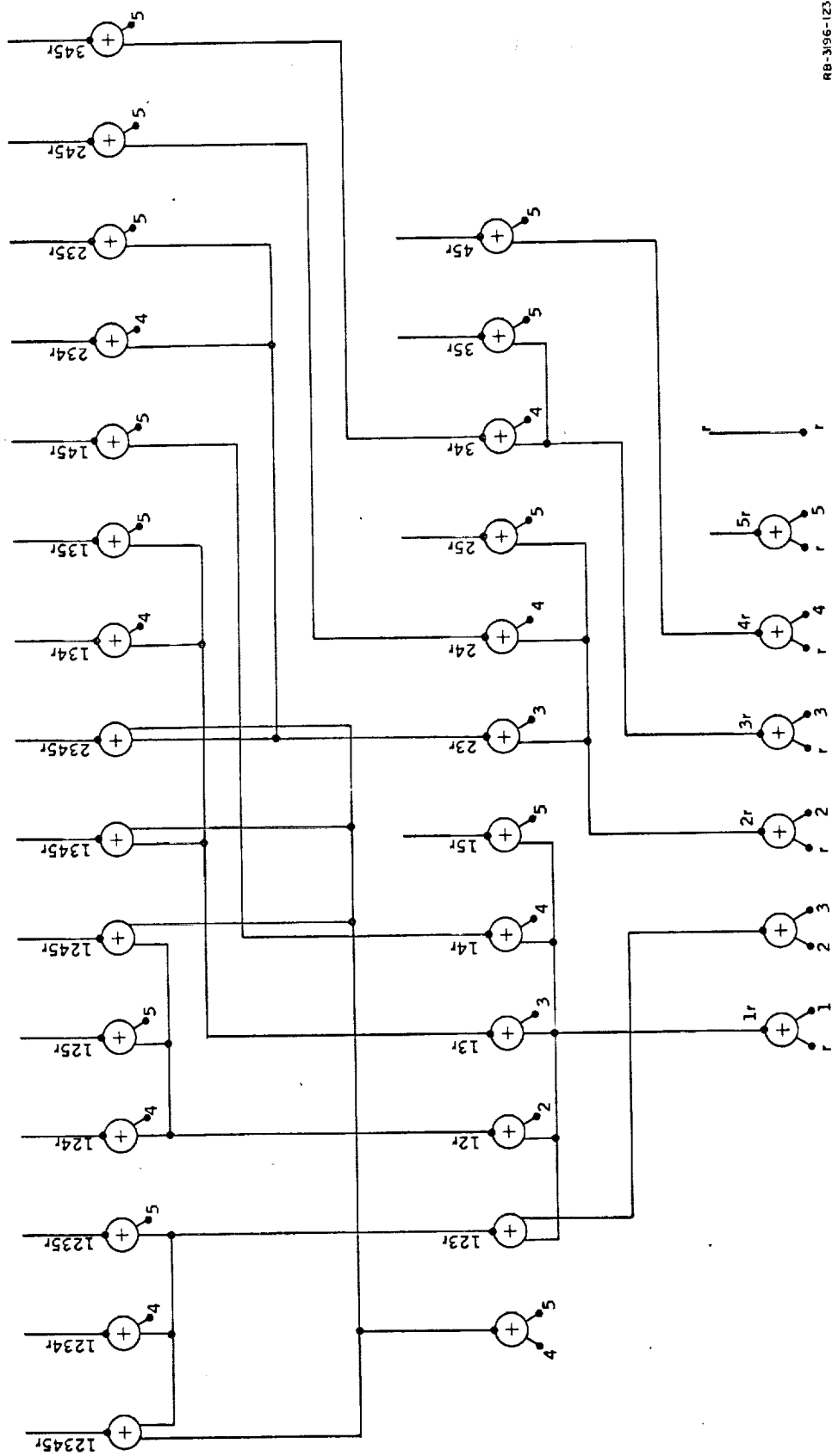
FIG. 2.8   MINIMUM-LEVEL LINEAR TREE, n = 5

Another type of linear tree which has practical importance is one for which all of the outputs occur at the same level. Such a tree would be required for instance if MAD-type two-input exclusive-OR gates were used, in order to deliver the recoder outputs to the access switch at the same clock time. Trees of this type are called *minimum level synchronous linear trees*.

Table 2.2

PROPERTIES OF MINIMUM-LEVEL
LINEAR TREES

n	p	$2^n - 1$	NUMBER OF NON-OUTPUT ELEMENTS	TOTAL NUMBER OF ELEMENTS
1	1	1	0	1
2	2	3	0	3
3	2	7	1	8
4	3	15	1	16
5	3	31	2	33
6	3	63	5	68
7	3	127	12	139
8	4	255	12	267
9	4	511	13	524
10	4	1023	16	1039
11	4	2047	23	2070
12	4	4095	38	4133
13	4	8191	69	8260
14	4	16383	132	16515
15	4	32767	259	33026
16	5	65535	259	65794

In this case the additional restriction is imposed that all paths from active inputs to network outputs traverse the same number of elements. Active inputs are distinguished in this case since the possibility of null, or $\emptyset$ inputs must be admitted. These functionally convert the exclusive-OR elements into simple delay elements.

Again it is convenient first to consider the case for which $n = 2^p - 1$. Using $p = 2$ as an example, there must be $2^n = 8$ gates which form the final network outputs. As indicated in Fig. 2.9 it would be possible to realize each such output disjointly with a tree of $n = 3$ gates for each output. Furthermore, if the variables of each tree are ordered in the same way—that is, if the same external input of each tree is always either the same variable, or is marked $\emptyset$—a great deal of redundancy can be expected among the trees in the sense that many subtrees will be
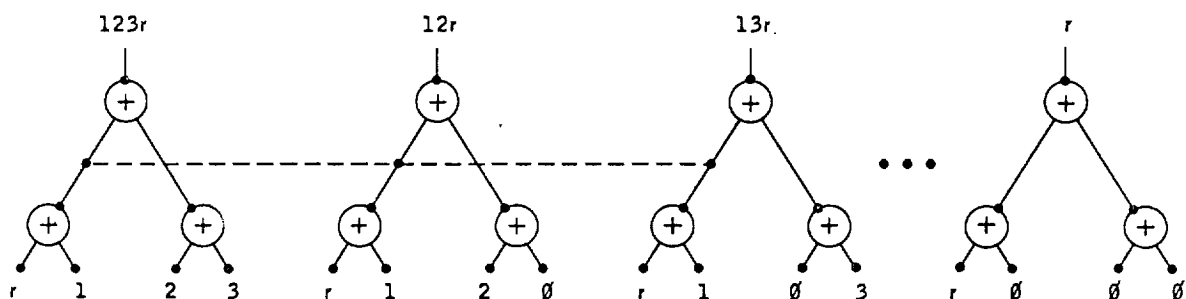
RB—3196—109

FIG. 2.9 A MINIMUM-LEVEL SYNCHRONOUS LINEAR-TREE REALIZATION FOR n = 3

65

devoted to realizing the same internal function. For example, the connection indicated by a dashed line in Fig. 2.9 can be made since the same function is formed at all three points. Hence, two of the three subnetworks can be eliminated. Although the over-all linear tree with this kind of redundancy eliminated is no longer composed of disjoint trees, the level requirement clearly is met.

Another kind of possible reduction is exemplified by the bottom-rightmost element of Fig. 2.9. Any element with both inputs marked $\emptyset$ can be eliminated provided its output, hence one of the inputs to the succeeding element, is marked $\emptyset$. This kind of simplification also does not disturb the level requirement since only inactive $\emptyset$ inputs are involved.

A systematic algorithm for constructing synchronous linear trees can be based on these observations: First, the required network outputs are realized with disjoint trees having identically ordered inputs, except for the null inputs where they are required, and second, the two kinds of redundancies noted are removed to form the final network.

For the case $n = 2^p - 1$, it is possible to enumerate the number of elements required by counting those required in each level. In all cases, the output level, p, will have 2^n elements. Level $p - 1$ will have one element for each different function formed on that level; furthermore every possible function (except $\emptyset \oplus \emptyset \oplus \ldots \oplus \emptyset$) will be required by one of the original disjoint trees. Hence exactly

$$2^{(n-1)/2} + 2^{(n+1)/2} - 1$$

elements are required in level $p - 1$. Similarly, level $p - 2$ will require

$$2^{[(n+1)/4]-1} + 3[2^{(n+1)/4} - 1] \quad ,$$

and level $p - j$ will require

$$2^{2^{p-j}-1} + (2^j - 1)\left(2^{2^{p-j}} - 1\right) \quad .$$

Summing over all levels,

$$N_n \le \sum_{j=0}^{p-1} \left[2^{2^{p-j}-1} + (2^j - 1)\left(2^{2^{p-j}} - 1\right) \right]$$

66

or

$$N_n \leq 2^p - p - 1 + \sum_{j=0}^{p-1} 2^{2^{p-j}-1}(2^{j+1} - 1) \qquad (2.26)$$

is an upper bound on the number of elements required.

A similar upper bound for $n = m + 2^{p-1} - 1$ can be obtained under the assumption that the 2^p first-level inputs to the original disjoint trees are labelled in the order r, x_1, x_2, ..., x_n, $\emptyset$, $\emptyset$, $\emptyset$, ..., $\emptyset$, where enough $\emptyset$'s (namely $2^{p-1} - m$) are appended to label each input. This does not seem interesting, however, since the number of elements required varies depending on just where in the listing the $\emptyset$'s are distributed. The orderings that for this construction result in the minimum number of elements are listed for $n \leq 10$ in Table 2.3.[†]

Table 2.3

PROPERTIES OF MINIMUM-LEVEL SYNCHRONOUS LINEAR TREES

n	p	VARIABLE ORDERING	N_n
1	1	01	2
2	2	012∅	7
3	2	0123	13
4	3	012∅3∅4∅	28
5	3	012∅345∅	50
6	3	0123456∅	88
7	3	01234567	162
8	4	012∅3∅4∅5∅6∅7∅8∅	309
9	4	012∅345∅6∅7∅8∅9∅	587
10	4	012∅345∅678∅9∅10∅	1120

Examples of minimum-level synchronous linear trees for $n = 2$, 3, 4, and 5 are shown in Figs. 2.10 and 2.11.

D. LATIN SQUARE SWITCH RECODERS

An entirely different type of reliable access switch that operates from a dense address code is a two-dimensional square coordinate switch with Latin-square redundancy.[15] Such a switch has two coordinate sets and s redundant sets of m wires each, where $s \leq m_0 - 1$ and m_0 is the least prime power in the factorization of m. There are m^2 outputs; hence, to use this type of switch to decode a dense address register code, it is necessary that $m = 2^k$. From this it follows that the number of outputs is 2^{2k}; therefore, an even number of address register bits, $n = 2k$, must be used. As the least prime power in the factorization of $m = 2^k$ is 2^k, $s = 2^k - 1$ mutually orthogonal redundant sets of wires may be threaded in addition to the two coordinate sets, yielding a zero-noise load-sharing read-access switch. For convenience, read-access switches of this type will be called L switches.

---

[*] Minimum-level linear trees and minimum-level synchronous linear trees were developed and analyzed by Robert A. Short.
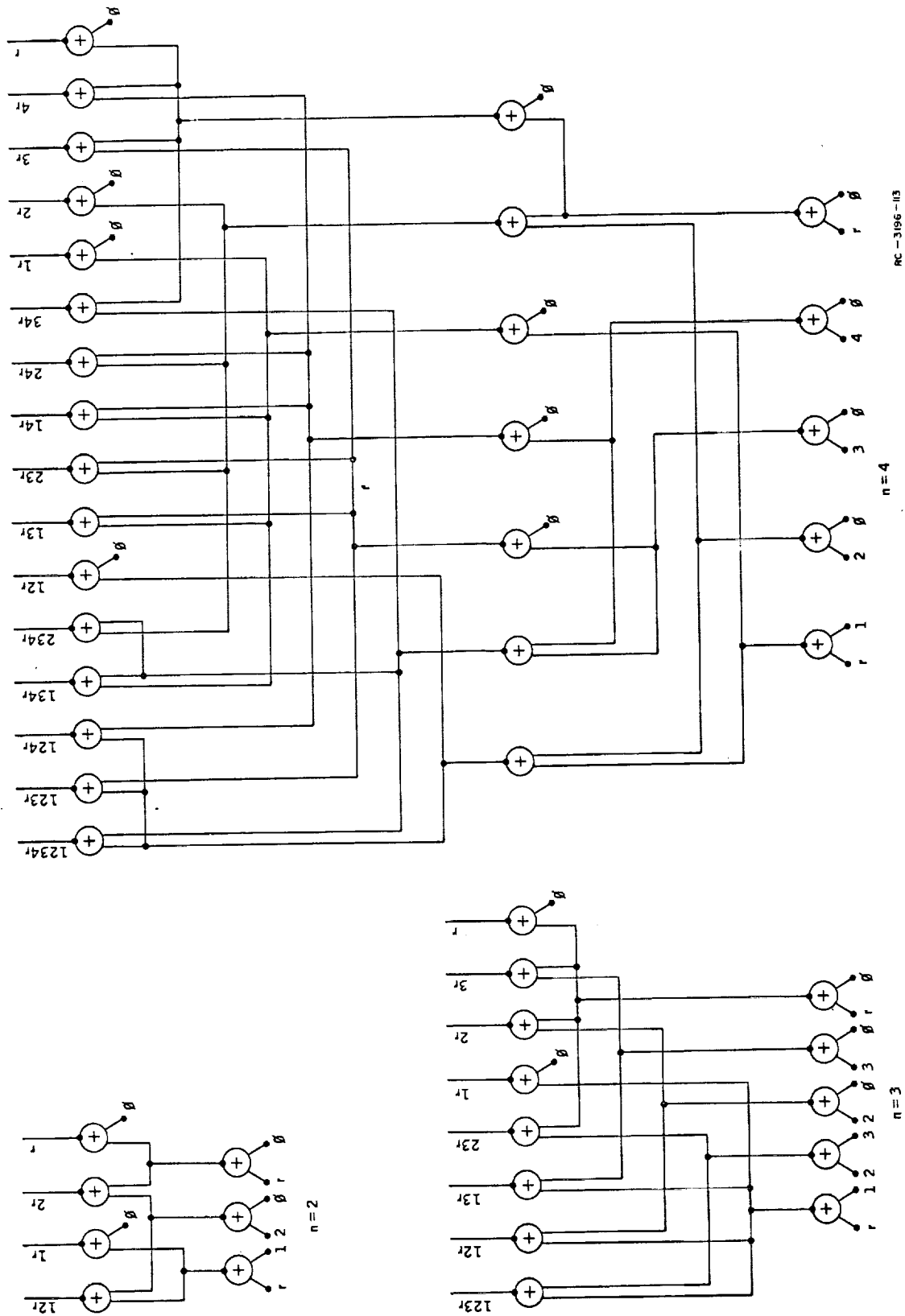
FIG. 2.10   MINIMUM-LEVEL SYNCHRONOUS LINEAR TREES FOR n = 2, 3, 4

RC-3196-113
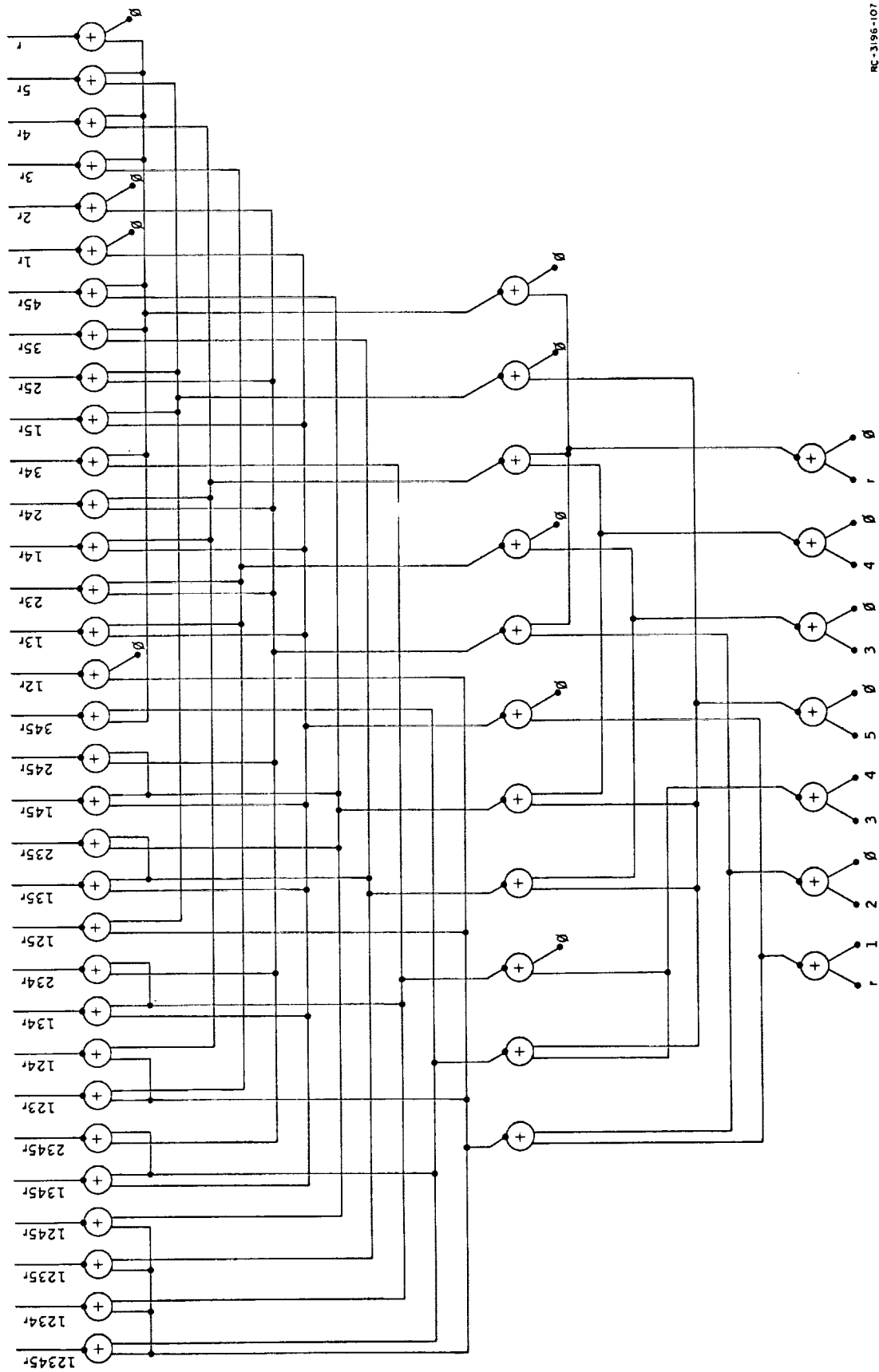
FIG. 2.11 MINIMUM-LEVEL SYNCHRONOUS LINEAR TREE FOR $n = 5$ (ϕ = False Input)

There probably are no more than four L switches worthy of practical consideration: these are the switches for $n = 2$, 4, 6, and 8. Only the first two of these will be considered in this report.

The winding, read-selection, and write-selection matrices for a four-output (2×2) L switch are found[15] to be

$$\underline{W}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & 1 & 0 & 1 & -1 & -1 & -1 \\ 0 & 1 & 1 & 0 & 0 & 1 & -1 & -1 & -1 \\ 0 & 1 & 0 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix} \quad ,$$

$$\underline{S}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \end{matrix} \quad , \qquad \underline{S}_2^* = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \end{matrix} \quad . \qquad (2.27)$$

z_1:	0 0 1 1	z_1:	0 0 1 1
z_2:	0 1 0 1	z_2:	0 1 0 1

Multiplying the matrices of Eq. (2.27) to form the read and write excitation matrices,

$$\underline{X}_2 = \underline{W}_2 \underline{S}_2 = 2\underline{I}_2 \ , \qquad \underline{X}_2^* = \underline{W}_2 \underline{S}_2^* = -2\underline{I}_2 \qquad (2.28)$$

where $\underline{I}_2$ is the 4×4 unit matrix. Thus, as predicted, the L switch of Eq. (2.27) is a zero-noise switch with a load-sharing factor of two. The nine outputs d_1, d_2, ..., d_9 of the recoder for this L switch may be written in terms of the two address register bits, z_1 and z_2, by inspection of Eq. (2.27), as follows:

$$d_1 = r \oplus z_1 \ , \quad d_2 = r \oplus z_1' \ , \quad d_3 = r \oplus z_2 \ , \quad d_4 = r \oplus z_2' \ ,$$

$$d_5 = r \oplus z_1 \oplus z_2 \ , \quad d_6 = r \oplus z_1 \oplus z_2' \ , \quad d_7 = 1 \ , \quad d_8 = d_9 = r' \ . \quad (2.29)$$

This result should be compared with the four-output C-switch recoder of Eq. (2.24). It is seen that the linear trees that were developed for the four-output C switch serve for the four-output L switch.

Next, the logical design will be considered of a recoder for the 16-output L switch. This switch has the parameters $m = 4$, $n = 2$; the winding matrix is found to be[15]

$$
\underline{W}_4 = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & -1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & -1
\end{bmatrix} , \quad (2.30)
$$

the read-selection matrix is found to be

$$
\underline{S}_4 =
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{matrix}
d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \\ d_{16} \\ d_{17} \\ d_{18} \\ d_{19} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \\ d_{24}
\end{matrix}
\quad, \qquad (2.31)
$$

z_1:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
z_2:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
z_3:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
z_4:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
t_1:	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
t_2:	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
t_3:	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
t_4:	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
t_5:	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
t_6:	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1

and the write-selection matrix is found to be

$$
\underline{S}^{*}_{4} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
. \qquad (2.32)
$$

By direct multiplication of the matrices in Eqs. (2.30), (2.31), and (2.32), the read and write excitation matrices are found to be

$$
\underline{X}_{4} = \underline{W}_{4}\underline{S}_{4} = 4\underline{I}_{4} \quad ,
$$

$$
\underline{X}^{*}_{4} = \underline{W}_{4}\underline{S}^{*}_{4} = -4\underline{I}_{4} \qquad (2.33)
$$

where $\underline{I}_{4}$ is the 16 × 16 unit matrix. Therefore, this L switch is of the zero noise variety, and it has a load-sharing factor of four.

By analogy with the methods used to develop the read portion of the logic for the C-switch recoder [Eq. (2.20)], the twenty-four outputs d_1, d_2, ..., d_{24} of the L-switch recoder may be listed beside the read-selection matrix, as shown in Eq. (2.31). The four address-register bits, z_1, z_2, z_3 and z_4, are listed at the bottom of the matrix. Then from Eq. (2.31) the read logic for the first eight outputs, which correspond to the coordinate portion of the access switch, is

$$_1d_1 = z_1'z_2' , \quad _1d_2 = z_1'z_2 , \quad _1d_3 = z_1z_2' , \quad _1d_4 = z_1z_2 ,$$

$$_1d_5 = z_3'z_4' , \quad _1d_6 = z_3'z_4 , \quad _1d_7 = z_3z_4' , \quad _1d_8 = z_3z_4 . \quad (2.34)$$

The presubscript 1 indicates that Eq. (2.34) refers only to the read portion of the logic. The logic for the remaining outputs is more complicated; therefore, it is convenient to introduce six intermediate variables, t_i, arranged so that the remaining outputs are similar to Eq. (2.34). That is, the t_i of Eq. (2.31) are defined by

$$_1d_9 = t_1't_2' , \quad _1d_{10} = t_1't_2 , \quad _1d_{11} = t_1t_2' , \quad _1d_{12} = t_1t_2 ,$$

$$_1d_{13} = t_3't_4' , \quad _1d_{14} = t_3't_4 , \quad _1d_{15} = t_3t_4' , \quad _1d_{16} = t_3t_4 ,$$

$$_1d_{17} = t_5't_6' , \quad _1d_{18} = t_5't_6 , \quad _1d_{19} = t_5t_6' , \quad _1d_{20} = t_5t_6 . \quad (2.35)$$

And the bias outputs are the trivial functions

$$_1d_{21} = 1 , \quad _1d_{22} = _1d_{23} = _1d_{24} = 0 . \quad (2.36)$$

The auxiliary variables t_i in Eq. (2.31) may be determined in terms of the z_i by conventional logical methods. The results are

$$t_1 = z_1 \oplus z_3 , \quad t_2 = z_2 \oplus z_4 , \quad t_3 = z_1 \oplus z_2 \oplus z_3 ,$$

$$t_4 = z_1 \oplus z_4 , \quad t_5 = z_2 \oplus z_3 , \quad t_6 = z_1 \oplus z_2 \oplus z_4 . \quad (2.37)$$

Generalizing these results to include both the read and write operations, an examination of the write-selection matrix, Eq. (2.32), indicates

74

that all drivers that were not used in the read operation are energized. Therefore, Eqs. (2.34), (2.35), and (2.36) become

$$d_i = r \oplus {}_1 d_i' \ , \qquad i = 1, 2, \ldots, 20$$

$$d_{21} = 1$$

$$d_{22} = d_{23} = d_{24} = r' \ . \tag{2.38}$$

Equations (2.38), (2.34), (2.35), and (2.37) define the complete logic for the sixteen-output L-switch recoder.

While the four-output L-switch recoder equations, Eq. (2.29), were essentially the same as the recoder equations for the four-output C switch, Eq. (2.24), the sixteen-output C-switch and L-switch recoder equations are significantly different. From Eqs. (2.38), (2.34), (2.35), and (2.37), it is seen that this recoder consists of a partial linear tree, which produces the t_i, followed by one level of AND-OR-gates to produce the ${}_1 d_i$ and ${}_1 d_i'$, and finally by one level of exclusive-OR gates to produce the d_i. This tree, which is shown as Fig. 2.12, is most conveniently drawn by assuming as before that each two-variable exclusive-OR gate has both the variables and their complements available, and produces both the desired function and the complement. Similarly, on this figure a single level of AND-OR-gates, which produces the four maxterms of the two input variables, is assumed to have complementary inputs and outputs. By obvious modifications of the input portions in Fig. 2.12, a synchronous L-switch recoder may be drawn.

A comparison of Fig. 2.12 with Figs. 2.4, 2.7, and 2.10 indicates that L-switch recoders are somewhat more complicated than C-switch recoders. Using techniques similar to those which led to the sixteen-output L switch and its recoder, equations may be developed for larger L switches and their recoders. However, in view of the switches to be considered in the next section, it does not seem to be of great practical importance to do this.

E. SOBELMAN SWITCH RECODERS

The sixteen-output L switch and the corresponding recoder considered in the previous section were by no means unique. If any row or any column

77
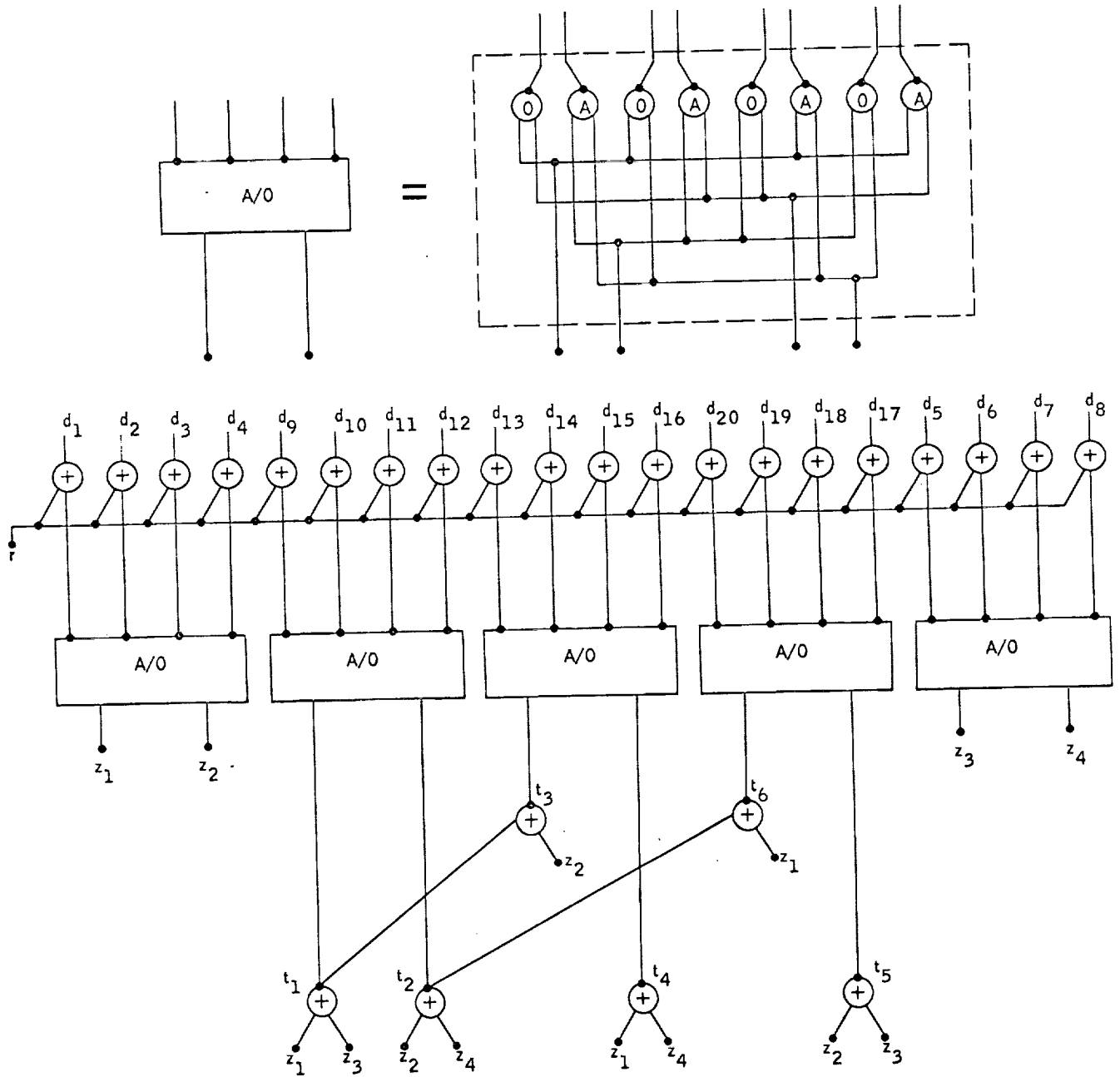
FIG. 2.12   RECODER FOR SIXTEEN-OUTPUT L SWITCH

RB-3196-122

rearrangement is made in the L-switch winding matrix, Eq. (2.30), this means only that the access-switch input and output wires are correspondingly renamed. However, such row and column rearrangements do affect the recoder logic. That this is so can be seen by reference to the read-selection matrix for the sixteen-output L switch, Eq. (2.31). If a particular row rearrangement is made in $\underline{W}_4$, then the same rearrangement must be applied to the columns of $\underline{S}_4$. Similarly, a column rearrangement in $\underline{W}_4$ calls for a corresponding row rearrangement in $\underline{S}_4$. Recalling the manner in which the auxiliary recoder functions t_i were introduced in Eq. (2.31), it should be evident that row and column rearrangements change these functions.

From this discussion it should be clear that a study of the row and column rearrangements of L switches might be made, with the purpose of finding economical recoders. In particular, if the following rearrangement is made of $\underline{S}_4$ in Eq. (2.31):

Rows: interchange 10 and 12, 14 and 16,

Columns: order 0, 6, 5, 3, 8, 14, 13, 11, 2, 4, 7, 1, 10, 12, 15, 9,

then Eq. (2.31) becomes

$$
\underline{S}_4 = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{matrix}
d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \\ d_{16} \\ d_{17} \\ d_{18} \\ d_{19} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \\ d_{24}
\end{matrix}
\qquad (2.39)
$$

z_1:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
z_2:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
z_3:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
z_4:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
s_1:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
s_2:	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
s_3:	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
s_4:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
s_5:	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
s_6:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
s_7:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
s_8:	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
s_9:	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
s_{10}:	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

Developing the recoder logic for Eq. (2.39) in a manner analogous to that for Eq. (2.31), ten auxiliary functions, s_1, s_2, ..., s_{10}, are defined in Eq. (2.39) in such a way that

$$_1d_1 = s_1's_2' \, , \quad _1d_2 = s_1's_2 \, , \quad _1d_3 = s_1s_2' \, , \quad _1d_4 = s_1s_2 \, ,$$

$$_1d_5 = s_3's_4' \, , \quad _1d_6 = s_3's_4 \, , \quad _1d_7 = s_3s_4' \, , \quad _1d_8 = s_3s_4 \, ,$$

$$_1d_9 = s_5's_6' \, , \quad _1d_{10} = s_5's_6 \, , \quad _1d_{11} = s_5s_6' \, , \quad _1d_{12} = s_5s_6 \, ,$$

$$_1d_{13} = s_7's_8' \, , \quad _1d_{14} = s_7's_8 \, , \quad _1d_{15} = s_7s_8' \, , \quad _1d_{16} = s_7s_8 \, ,$$

$$_1d_{17} = s_9's_{10}' \, , \quad _1d_{18} = s_9's_{10} \, , \quad _1d_{19} = s_9s_{10}' \, , \quad _1d_{20} = s_9s_{10} \, . \quad (2.40)$$

The auxiliary functions, s_i, in Eq. (2.39) may be determined in terms of the address register variables by conventional logical methods. The results are

$$s_1 = z_2 \, , \quad s_2 = z_3 \oplus z_4 \, , \quad s_3 = z_1 \oplus z_4 \, , \quad s_4 = z_3 \, ,$$

$$s_5 = z_1 \oplus z_2 \, , \quad s_6 = z_4 \, , \quad s_7 = z_1 \, , \quad s_8 = z_2 \oplus z_3 \, ,$$

$$s_9 = z_1 \oplus z_3 \, , \quad s_{10} = z_2 \oplus z_4 \, . \quad (2.41)$$

By analogy with Eq. (2.38), when both the read and write operations are considered, the recoder outputs, d_i, are

$$d_i = r \oplus {}_1d_2' \qquad i = 1, 2, \ldots, 20$$

$$d_{21} = 1$$

$$d_{22} = d_{23} = d_{24} = r' \, . \quad (2.42)$$

The recoder logic represented by Eqs. (2.40), (2.41), and (2.42) is drawn as Fig. 2.13. Comparing Figs. 2.12 and 2.13, it is seen that while the same amount of equipment is needed for this revised recoder, the number of logical levels is reduced.

If one had to investigate all row and column rearrangements of the L-switch winding matrix to find an efficient recoder, the problem would
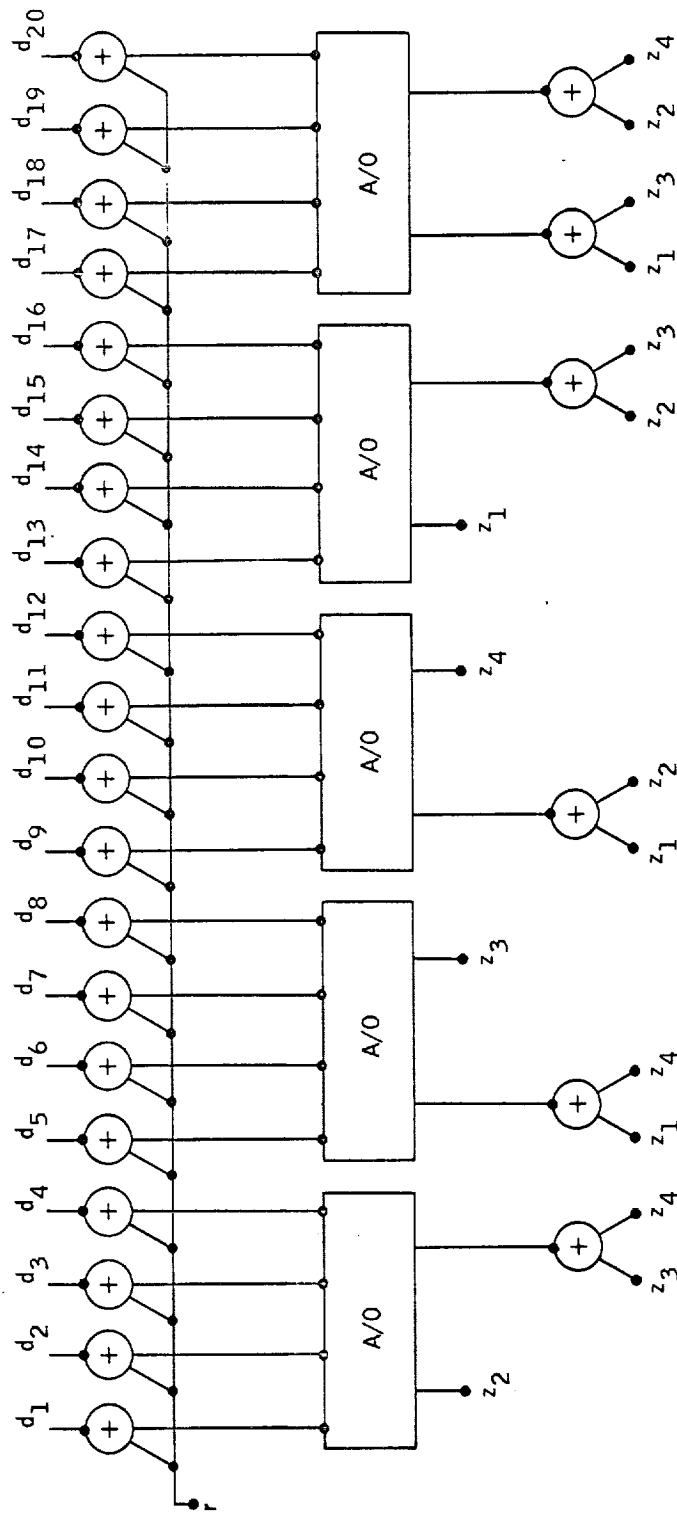
FIG. 2.13   RECODER FOR SIXTEEN-OUTPUT S SWITCH

RB-3196-121

be formidable indeed. Fortunately, one particular rearrangement can be found by using a method due to Sobelman.[16] Sobelman has developed a graphical method for finding the winding pattern and decoding for a certain class of multiple coincidence memory (not access switch). His approach is to find p sets of wires to thread the memory cores in such a way that each core in every memory plane is intersected by one wire from each set. A requirement on the threading is that if one wire in each set is energized, one core receives p units of magnetomotive force, while all others receive zero or one unit. While other workers[17] have approached this same problem by starting with two coordinate sets of wires, which correspond to a rectangular grid, and have added redundant sets in such a way that each added set obeys the required properties, Sobelman has considered the problem differently. He has eliminated the coordinate sets of wires, and has instead looked for sets of wires that have a particular decoding property—namely, a linear tree followed by AND-OR-gates.

Just as it has been possible in the past work on access techniques to convert the methods for wiring multiple-coincidence memory planes into the design of access switches, Sobelman's algorithm may be so adapted. When this is done it is found that such switches, which will be termed $\underline{S}$ switches, are row and column rearrangements of L switches, and consequently are restricted to 2^{2k} outputs, where $n = 2k$ address register bits are employed. Furthermore, except for the four-output and sixteen-output S switches, they are load-sharing but not zero noise; an examination of Sobelman's algorithm shows that with his assumptions they cannot be. However, a recognition of the relationship between S switches and L switches allows one to form a zero-noise load-sharing S switch for the sixty-four-output case by starting with Sobelman's decoding method and adding to it in an appropriate way.

As these methods for generating a zero-noise sixty-four-output S switch are somewhat ad hoc, only the results will be given here. For the purpose of condensation, the unaugmented $\underline{R}_8$ matrix[3] will be given. To find the read-selection matrix, $\underline{S}_8$, each row of $\underline{R}_8^T$ is replaced with eight rows, and a 1 is placed in the kth of these eight rows for each digit k in $\underline{R}_8^T$. Finally, appropriate rows for the bias drivers are appended to the bottom of $\underline{S}_8$. This $\underline{R}_8$ matrix is shown as Fig. 2.14. The columns to the right of the partition are the added columns to give the S switch a zero-noise property.

Figure 2.14 contains the recoder equations for this S switch. An attempt to design a sixty-four-output L-switch recoder has resulted in such a complicated logic that it is deemed infeasible to record it here. It is sufficient to say that the sixty-four-output S-switch recoder indicated in Fig. 2.14 is substantially simpler than the corresponding L-switch recoder.

The four-output S switch has a recoding logic essentially the same as the corresponding L switch. The sixteen-output S switch and its recoder were described by Eqs. (2.39), (2.40), (2.41), and (2.42), as well as by Fig. 2.13, while the sixty-four-output S switch and its recoder were described by Fig. 2.14. The next size S switch, and probably the only remaining one of practical importance, is the one having 256 outputs. Sobelman's decoding for this switch consists of seven wire sets; hence, it is necessary to find ten additional sets to produce a zero-noise switch having this number of outputs. This has not been done; however, it is probable that it can be done if an appropriate amount of time is devoted to the problem.

```
0 0 0 0 0 0 0 0
1 1 4 2 1 1 0 5 7
1 2 2 1 0 2 4 6 3
0 3 6 3 1 5 3 3 3
1 2 2 1 0 2 4 6 3
2 1 7 3 3 4 6 7 0
3 0 3 1 2 5 0 4 6
3 5 2 3 5 2 7 1 1
2 2 1 0 2 1 4 6 3
2 4 2 4 1 0 1 2 5
1 7 3 3 4 6 7 0 0
3 6 0 5 1 4 3 6 1
3 5 6 6 0 1 1 7 2
0 6 3 4 3 1 5 4 4
1 4 1 5 3 5 7 0 2
1 7 7 6 2 0 5 3 1
7 6 2 0 5 3 1 4 4
6 3 4 3 1 5 4 5 5
7 2 0 5 3 1 4 0 6
6 0 5 1 4 3 6 1 2
6 1 3 7 0 6 5 7 3
7 2 6 0 1 5 4 6 6
4 3 0 6 7 6 3 7 7
5 4 2 3 5 7 6 7 0
4 2 6 1 3 7 0 6 4
5 2 7 6 4 0 2 3 3
4 6 3 2 3 6 1 5 4
6 3 4 5 2 7 1 4 0
7 2 1 1 7 4 0 2 6
7 1 6 4 3 3 6 3 7
6 4 2 3 3 0 6 0 1
6 2 3 3 5 6 0 6 4
7 1 1 6 4 0 4 4 0
7 5 6 1 1 7 4 5 7
4 3 3 0 7 4 5 1 0
7 3 0 7 4 5 1 4 6
5 0 1 2 6 4 2 2 7
6 1 2 6 3 5 2 3 6
6 6 0 3 6 3 5 5 5
4 7 6 1 3 7 0 6 5
5 6 2 3 0 6 1 2 2
4 2 3 3 5 6 0 3 3
6 0 2 6 0 1 2 3 6
7 6 4 2 3 3 6 1 0
7 1 6 4 3 3 0 4 6
0 4 2 7 1 4 5 2 5
5 3 7 4 0 2 7 1 6
3 7 4 0 2 7 1 5 7
4 0 2 7 1 5 7 6 0
2 6 1 3 5 0 6 3 4
3 4 5 2 7 1 4 5 3
6 5 2 7 1 4 0 2 7
3 4 7 1 0 3 2 7 6
4 5 1 4 0 7 3 1 7
5 2 7 5 6 0 1 0 0
7 1 0 6 7 4 5 6 4
0 7 4 5 1 0 2 6 2
4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4
```

$d_{i1} = r \oplus d'_{i1}$

$d'_{i1} = s'_{i1} s'_{i2} s'_{i3}$, $d'_{i2} = s'_{i1} s'_{i2} s'_{i3}$

$\cdots$, $d'_{i8} = s'_{i1} s'_{i2} s'_{i3}$, $i = 1, 2, \ldots, 9$

$s_{11} = z_1 \oplus z_2$, $s_{12} = z_3 \oplus z_4$, $s_{13} = z_5 \oplus z_6$

$s_{21} = z_2 \oplus z_3$, $s_{22} = z_4 \oplus z_5$, $s_{23} = z_6$

$s_{31} = z_1 \oplus z_6$, $s_{32} = z_3 \oplus z_5$, $s_{33} = z_4$

$s_{41} = z_1 \oplus z_3$, $s_{42} = z_2 \oplus z_6$, $s_{43} = z_5$

$s_{51} = z_1$, $s_{52} = z_2 \oplus z_4$, $s_{53} = z_3 \oplus z_6$

$s_{61} = z_1 \oplus z_5$, $s_{62} = z_2$, $s_{63} = z_4 \oplus z_6$

$s_{71} = z_1 \oplus z_4$, $s_{72} = z_2 \oplus z_5$, $s_{73} = z_3$

$s_{81} = z_1 \oplus z_3$, $s_{82} = z_2 \oplus z_6$, $s_{83} = z_1 \oplus z_4 \oplus z_5$

$s_{91} = z_1 \oplus z_5 \oplus z_6$, $s_{92} = z_1 \oplus z_2 \oplus z_3$, $s_{93} = z_3 \oplus z_4 \oplus z_6$

FIG. 2.14 64-OUTPUT S SWITCH

III EXAMPLES OF READ-ACCESS SWITCH RECODERS

A. MAD RECODERS

While the logical equations for several types of read-access switch recoders have been developed in the previous sections, it appears to be of some practical interest at least to suggest certain methods for implementing these equations in hardware. As the fundamental purpose of this research is to develop reliable memory systems, it is natural to recommend that a maximum use be made of magnetic circuits and that the number of semi-conductors be kept as low as possible.

In this connection, it should be obvious that a magnetic multi-apertured device, or MAD, implementation should be considered for the two-input exclusive-OR gates that occur in all the previously discussed recoders. In using MADs for these recoders it should be noted that the r and w (which has been called r') signals are not truly complementary. While truly complementary signals x and y are related by $x \oplus y = 1$, r and w actually have the relationship $(rw)' = 1$; that is, both may be absent if the memory is idle, or either one may be true, but both may not be true. This means that for the case $r = w = 0$, or the quiescent memory case, the exclusive-OR gate that has r as an input must produce a false output. It happens that the two-level AND-OR logic that is used for the diode implementation of the exclusive-OR gates operates as described in this case. However, in a MAD implementation it appears that this may not be so. Therefore, in designing multiapertured magnetic recoders, special consideration must be given to the exclusive-OR gates that have r as one input; if necessary these can be replaced by two levels of AND-OR magnetic logic as was done in the diode case.

B. A COMMERCIAL C SWITCH AND RECODER

In order to have a basis for comparison, it seems reasonable to examine the circuitry for a commercially available C switch together with its recoder. The one chosen is from the IBM 7302 core storage,[18] which is used in the IBM 7090 Data Processing System. This memory consists of 16,384 words of 72 bits each, organized as a 128 × 128 × 72

coincident-current array. In each of the X and Y access dimensions there are 8 sixteen-output C switches; therefore, in each dimension three address bits correspond to segment gating, and four bits are used to specify one of the sixteen C-switch outputs.

On Fig. 2.15, one of the sixteen C switches required in this memory is drawn, together with its recoder. The circuit shown has been idealized by omitting 23 buffers. The recoder of Fig. 2.15 should be compared with the $n = 4$ linear trees shown in Figs. 2.4, 2.7, and 2.10. In particular, it is seen that an $n = 4$ minimum level synchronous linear tree requires no more than 28 exclusive-OR gates, while the circuit of Fig. 2.15 requires 20 exclusive-OR gates and 23 buffer stages. Hence, it is concluded that the results of the study of linear trees can be used to improve existing C-switch recoder logical designs.

C. S SWITCH WITH COMBINED DIODE-MAGNETIC RECODER

On Fig. 2.16 an S switch has been drawn for purpose of comparison with the C switch of Fig. 2.15. While both switches have sixteen outputs and are zero noise, the similarity stops there. The C switch has 32 drivers and a load-sharing factor of 16, while the S switch has 24 drivers and a load-sharing factor of 4. The C switch has 512 input windings while the S switch has only 144.

Thus, the sixteen-output S switch has fewer drivers and fewer windings than the corresponding C switch, while it has a smaller load-sharing factor.

The recoder logic for the S switch of Fig. 2.16 is given as Eqs. (2.40), (2.41), and (2.42). One way to implement this logic is to form the exclusive-ORs of Eq. (2.41) in a diode recoder, and then to form the two remaining equations magnetically. To illustrate the magnetic logic, it is necessary to consider the recoder output d_3. From Eqs. (2.40) and (2.42):
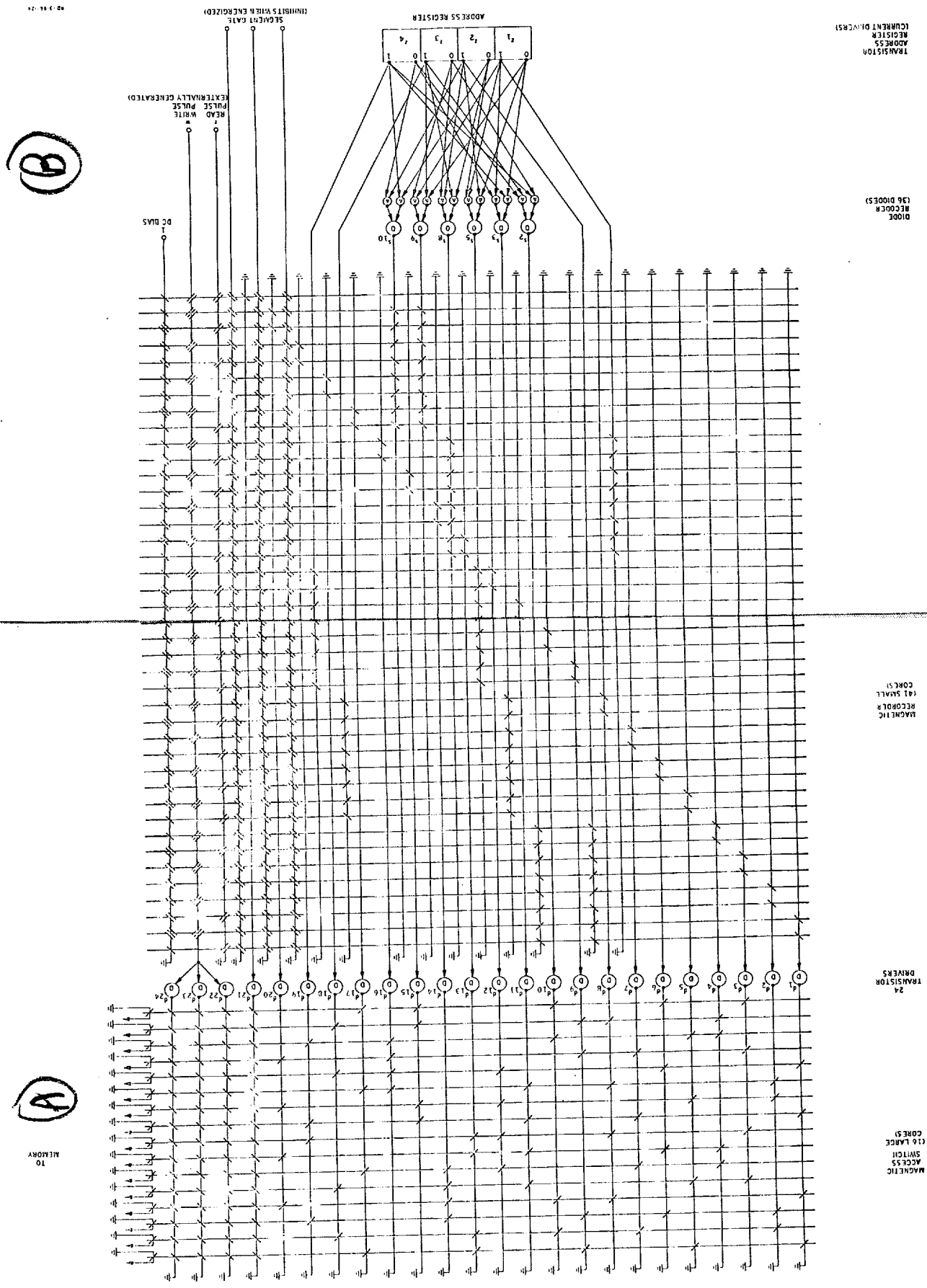
$$d_3 = r \oplus {}_1d_3' = r \oplus (s_1' + s_2) \quad ,$$

$$= rs_1s_2' + w(s_1' + s_2) \tag{2.43}$$

where the inputs, s_1 and s_2, are available either from the address register or as non-complementary outputs of diode exclusive-OR gates which

FIG. 2.15   C SWITCH AND RECODER USED IN THE 7090 COMPUTER

TRANSISTOR ADDRESS REGISTER (CURRENT DRIVERS)

DIODE RECODER (36 DIODES)

MAGNETIC RECODER (4) SMALL CORES)

24 TRANSISTOR DRIVERS

MAGNETIC ACCESS SWITCH (16 LARGE CORES)

ADDRESS REGISTER

SEGMENT GATE (INHIBITS WHEN ENERGIZED)

WRITE PULSE   READ PULSE (EXTERNALLY GENERATED)

DC BIAS

TO MEMORY

in turn are controlled by the bits of the address register. It is assumed here that the address register delivers *level* outputs—that is, these outputs can be considered to be constant for the duration of the memory cycle. The signals r and w are assumed to be pulses.

Equation (2.43) may be implemented magnetically in several ways. A particularly interesting method combines the principles of *linear-input logic* with those of *linear-output logic* to produce a *self-clocking network*. In a magnetic implementation of linear-input logic, the inputs are linearly combined by connecting them to a single magnetic toroid with windings having differing numbers of turns and polarities. In magnetic linear-output logic a single-output winding is connected in series with several toroids. The simplest form of linear-output logic is used here—namely, the implementation of the two-input inclusive-OR, which is simply a series connection of two toroids. One toroid produces the first term of Eq. (2.43) and the other produces the second term. Considering the first term of Eq. (2.43), a magnetic linear-input implementation is[†]

$$(s_2, \; 1 \; * \; s_1, \; r) \quad . \tag{2.44}$$

This implementation, however, does not take full advantage of the situation, for a reset signal is needed. The reset signal may be eliminated by the following logic:

$$(s_2, \; 2 \; * \; s_1, \; 2r) \quad . \tag{2.45}$$

An examination of Eq. (2.45) shows that during the quiescent period of the memory ($r = 0$) the toroid is held in the reset condition, regardless of the states of s_1 and s_2. When the write signal, r, occurs, it acts as the clocking pulse that switches the core providing $s_1 s_2' = 1$. When the r pulse is removed, the toroid is automatically reset. A network of this sort is termed *self-clocking*.

---

[*] The notation used here is described in an article by R. C. Minnick.[19] Briefly, each variable to the right of the asterisk is connected in the SET direction with the number of turns that corresponds to the coefficient of the variable, while each variable to the left of the asterisk is similarly connected in the RESET direction.

90

An examination of Eqs. (2.40) and (2.42) shows that there are just six essentially different linear-input functions to be produced. Using s_a and s_b to refer to any two of s_1, s_2, ..., s_{10}, these functions together with self-clocking linear-input logic implementation are listed in Table 2.4. Each of the functions d_1, d_2, ..., d_{20} is produced using two toroids with series-connected outputs; the logic for each of the first 40 toroids in the recoder of Fig. 2.16 is taken from Table 2.4. The functions d_{21}, d_{22}, d_{23}, and d_{24} are produced with one additional toroid in the recoder.

Table 2.4

LINEAR-INPUT IMPLEMENTATIONS

FUNCTION	IMPLEMENTATION
$rs_a's_b'$	$(s_a, s_b, 1 * 2r)$
$rs_a's_b$	$(s_a, 2 * s_b, 2r)$
rs_as_b	$(3 * s_a, s_b, 2r)$
$w(s_a + s_b)$	$(3 * s_a, s_b, 3w)$
$w(s_a + s_b')$	$(s_b, 2 * s_a, 3w)$
$w(s_a' + s_b')$	$(s_a, s_b, 1 * 3w)$

A further examination of Fig. 2.16 reveals the presence of three segment gate wires. When any one or more of these wires is energized, a sufficient magnetomotive force is applied to hold every toroid of the recoder in its reset state; this of course prevents the production of any output from the recoder. There is no limitation to the number of segment gate wires that can be used; three were chosen in Fig. 2.16 to make it comparable with Fig. 2.15. It should be noted that no additional toroids are needed for the segment gating in Fig. 2.15, while an eight-output decoding network (which is not shown in the figure) is necessary for the segment gating in the commercial C switch of Fig. 2.15.

D. S SWITCH WITH ALL-MAGNETIC RECODER

Other single-apertured magnetic recoders that combine linear-input logic, linear-output logic, and the self-clocking features exhibited in the network of Fig. 2.16 are possible. Illustrated in Fig. 2.17 is a possibility that does not use the diode recoder of Fig. 2.16 for the operation of the six exclusive-OR functions; furthermore, fewer toroids (18 versus 41) are required.

The reduced complexity of the recoder in Fig. 2.17 is due to the assignment of more of the logic to linear-output logic. That is, linear-input logic is used to decode the four-bit binary number N in the address register to set the Nth toroid upon application of r, and to reset the same Nth toroid upon application of w. The linear-output logic follows the same coding as the $\underline{W}$ matrix of the S switch. For instance, examination
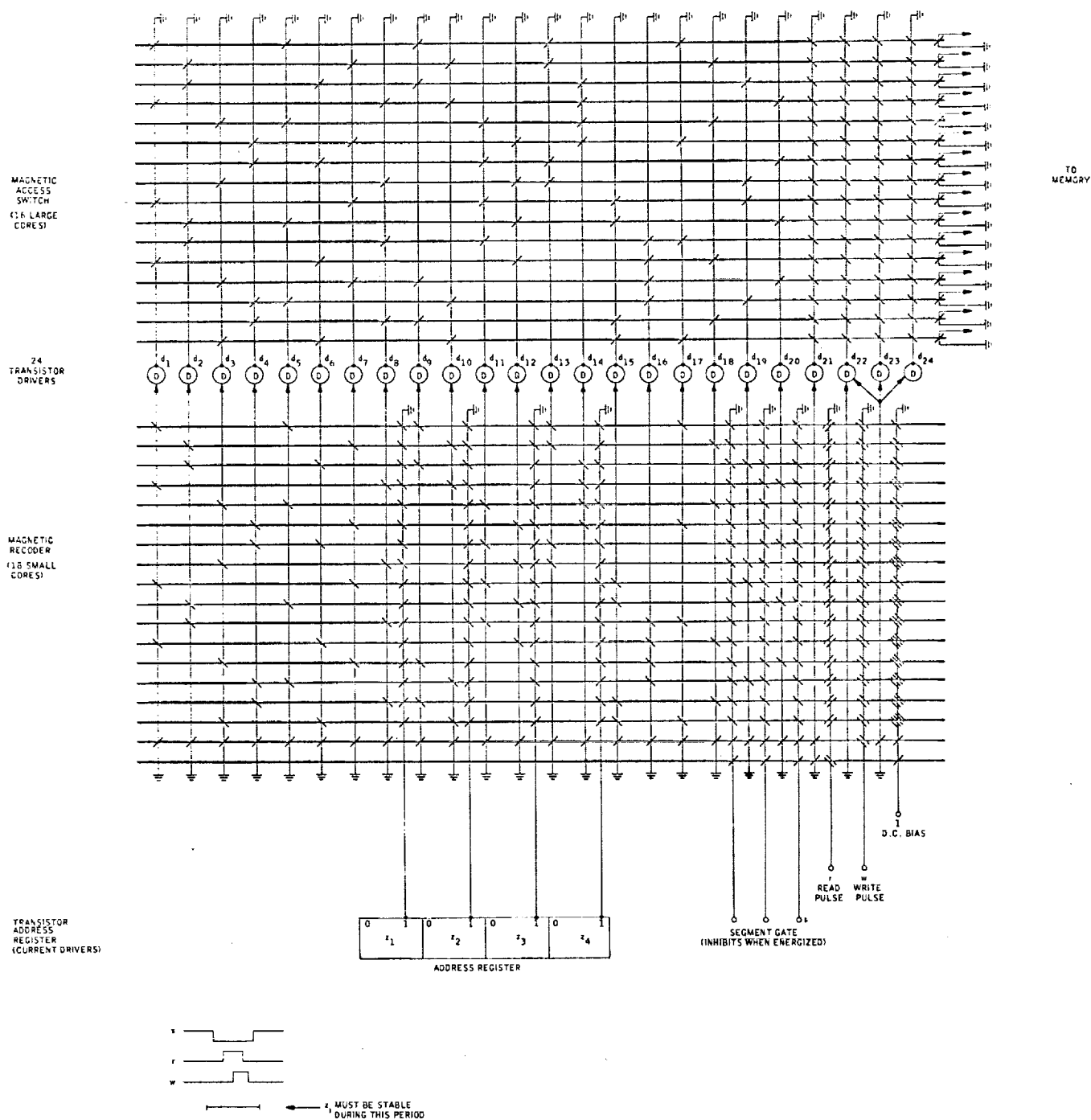
FIG. 2.17   S SWITCH WITH ALL-SINGLE-APERTURED MAGNETIC RECODER

of the recoder output d_3 shows that upon application of the read signal, r, d_3 is true provided the number in the address register is $N = 4$, 7, 12, or 15. But from Eqs. (2.40) and (2.41) the required read logic for the output d_3 is

$$_1 d_3 \;=\; r z_2 (z_2 + z_4)' \;\; , \tag{2.46}$$

which can be verified to be true when the number in the address register is $N = 4$, 7, 12, or 15. Indeed this must be the case, for the recoder equations were derived from the $\underline{W}$ matrix of the S switch.

The magnetic recoder of Fig. 2.17 is as economical as shown because the r and w pulses have been assumed to overlap. For this reason, immediately upon application of w, the magnetomotive forces due to r and to w cancel; this allows the toroid to be reset that was set when r was applied. The bottom two toroids in the recoder of Fig. 2.17 deliver appropriate thresholds for the linear-output windings. The segment gate windings operate similarly to those in Fig. 2.16.

One interesting comparison results between the recoders of Figs. 2.15 and 2.17; this is a count of the number of solder joints in the recoder portion of the switch. It appears that the diode recoder of Fig. 2.15 requires at least 376 solder joints, while the magnetic recoder of Fig. 2.17 requires only 34, or roughly 9% as many.
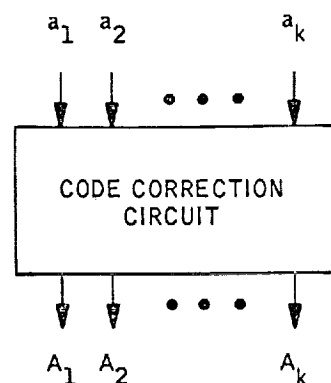
IV   CODE-CORRECTION CIRCUITS

A.   BLOCK CODE-CORRECTION CIRCUITS[†]

Referring to Fig. 2.18, the function of a code-correction circuit is to correct a word having the digits a_1, a_2, ..., a_k if no more than a certain maximum number of digits are in error. The corrected code word has the digits A_1, A_2, ..., A_k.

If the code words are represented in the form of some block code having a distance d, then a passive magnetic structure may be designed to effect this correction as follows. A magnetic core may be associated with each code word, except the word that contains all 0's, and a winding of ± 1 turn may connect each of the inputs a_i to the cores, corresponding to that digit being 1 or 0. A bias winding connects to each core with $(d - w - p)$ turns, where w is the *weight* or number of 1's in the code word, and p is some positive integer. It is known[20] that the minimum value of w in any code word is d, exclusive of the code word that contains all 0's. Hence, each core has $w \geq d$ positively connected windings, and in the absence of any faults of the inputs, the correct core is selected with a mmf of $w + d - w - p = d - p$ units. Considering next some core that should not be selected, it must have at least d different digits than the selected core. If this non-selected core has a digit 0 where the selected core has a digit 1, then its nominal select mmf of $d - p$ units is changed by -1 unit because an input wire connected to -1 turn is assured to carry current. On the other hand, if this non-selected core has a digit 1, where the selected core has a digit 0, its nominal select mmf is still changed by -1 unit because no input wire carries current to a winding of $+1$ turn. Hence, the

a_1 a_2 a_k

CODE CORRECTION CIRCUIT

A_1 A_2 A_k

AB—3196—III

FIG. 2.18   BLOCK DIAGRAM OF A CODE-CORRECTION CIRCUIT

---

[*] See also Part Three, Sec. II-E, for another application of this technique.

most positive mmf on any non-selected core, in the absence of errors in the input code, is $d - p - d = {}^-p$ units.

A set of k secondary windings connect to this corrector in such a way that for each core a one-turn winding is made to an output wire if the corresponding bit of that code should be a *1*, and zero turns are made if the corresponding bit should be *0*.

If an input code has r errors, the select mmf ranges by $\pm r$ units from its nominal value, as does each non-select mmf. In order to select the proper core under worst-case conditions, and not to select any other core, it is necessary that

$$d - p - r \geq 1 \quad , \tag{2.47}$$

$$^-p + r \leq 0 \quad . \tag{2.48}$$

One possible solution for this pair of inequalities is

$$r = p \leq (d - 1)/2 \quad . \tag{2.49}$$

If d is odd, the equality holds.

An example of a block code correction circuit for the (7,3) Hamming code of Eq. (2.12) is given as Fig. 2.19. As $d = 3$ for this code, $r = p = 1$ from Eq. (2.49); therefore, any single error is corrected by this code-correction circuit.

B. MULTIPLE-PARITY-CHECK CODE-CORRECTION CIRCUITS

Block code-correction circuits seem to be practical only for fairly small code sets, because for larger code sets the number of cores and the number of windings per core increase to prohibitively large values. For small code sets, the block code-correction circuits also have the disadvantage of a large redundancy ratio.

Another approach—one that appears to be more economical in redundancy ratio for small code sets, as well as in the number of cores and windings for large code sets—is based on multiple parity-check codes.[21] For these codes the non-redundant word is assumed to have n^2 bits,
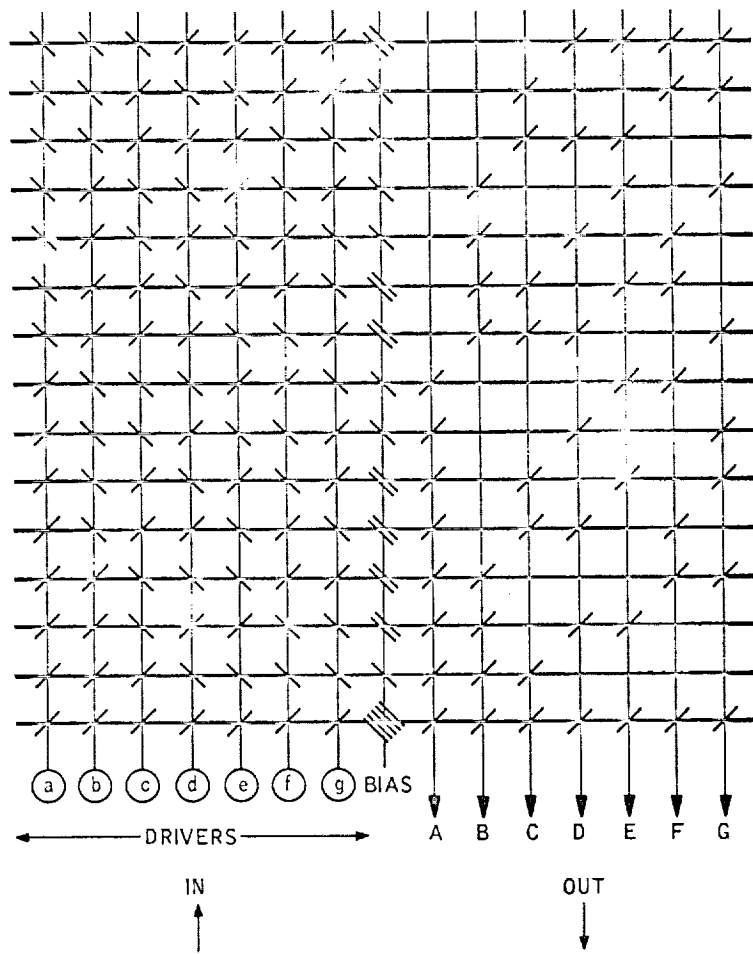
FIG. 2.19   CODE-CORRECTION CIRCUIT FOR THE (7,3) HAMMING CODE

RA-3196-117

a_{11}, a_{12}, ..., a_{nn}, which are organized as shown in Fig. 2.20. The redundant bits a_{0j} are defined in terms of vertical parity checks as
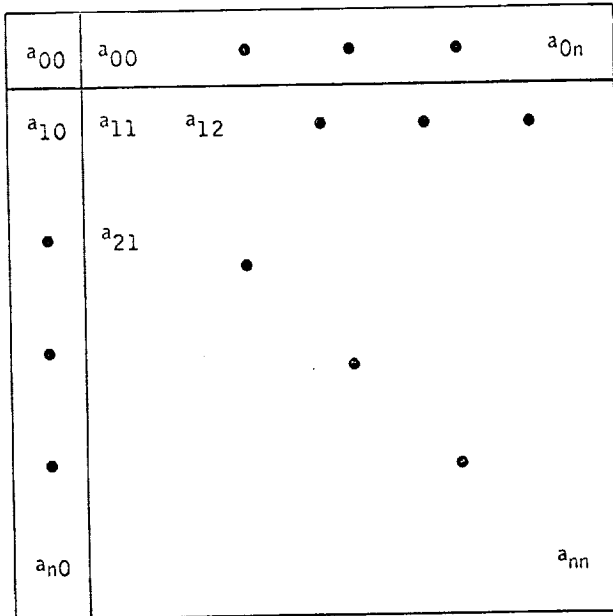
$$p_{vj} = \sum_{i=0}^{n} a_{ij} = 0 \qquad (2.50)$$

and similarly the redundant bits a_{i0} are defined by the horizontal parity checks

$$p_{hi} = \sum_{j=0}^{n} a_{ij} = 0 \quad . \qquad (2.51)$$

The redundant bit, a_{00}, is the normal parity bit over all n^2 non-redundant bits; it is defined as

RA-3196-110

FIG. 2.20   ORGANIZATION OF A
TWO-PARITY-CHECK
CODE

$$\sum_{j=0}^{n} a_{0j} = \sum_{i=0}^{n} a_{i0} = 0 \ . \ (2.52)$$

If there is only a single failure of one of the input bits, a_{ij}, then both the horizontal parity check and the vertical parity check through that bit will fail. Hence, the corrected output, A_{ij}, corresponding to an input bit, a_{ij}, is

$$A_{ij} = a_{ij} \oplus p_{hi} p_{vj} \quad . \qquad (2.53)$$

But from Eqs. (2.50) and (2.51),

$$p_{vj} = a_{ij} \oplus p_{vj}^{*}$$

$$p_{hi} = a_{ij} \oplus p_{hi}^{*} \qquad (2.54)$$

where   .

98

97

$$p_{vj}^* = a_{0j} \oplus a_{1j} \oplus \ldots \oplus a_{i-1,j} \oplus a_{i+1,j} \oplus \ldots \oplus a_{nj} \quad ,$$

$$p_{hi}^* = a_{i0} \oplus a_{i1} \oplus \ldots \oplus a_{i,j-1} \oplus a_{i,j+1} \oplus \ldots \oplus a_{in} \quad . \quad (2.55)$$

Combining Eqs. (2.54) and (2.55), and simplifying,

$$A_{ij} = a_{ij} \oplus [(a_{ij} \oplus p_{vj}^*)(a_{ij} \oplus p_{hi}^*)]$$

$$= a_{ij}p_{vj}^* + a_{ij}p_{hi}^* + p_{vj}^* p_{hi}^*$$

$$= Maj(a_{ij}, p_{vj}^*, p_{hi}^*) \quad (2.56)$$

From Eq. (2.56), it is evident that if it were possible with a simple magnetic circuit to generate the parity functions p_{vj}^* and p_{hi}^*, the majority function could be obtained with linear-output logic. There are a number of magnetic devices that are capable of generating the parity function; however, most of these are limited to two inputs. For fan-ins greater than two, the usual approach is to cascade several magnetic structures with appropriate coupling loops. Two structures for generating the parity function of more than two variables are those of Mina and Newhall,[22] and of Abbott and Suran.[23] The method of Ref. 22 works for any number of variables, n; however, 2^{n+1} holes are required. The method to be described requires only $3[(n + 1)/2] - 1$ holes (where [ ] indicates "the integer part of"). The method in the second reference is economical of holes, but it is limited to no more than four variables, and the output detector must sense a bipolar signal.

The structure in Fig. 2.21 has four input windings, labeled a, b, c, and d; these inputs are energized simultaneously with polarities that tend to switch flux downward in Legs 1, 2, 3, and 4. If any one of the four inputs is true, the flux in Leg 5 will be switched upward because this leg is in the shortest
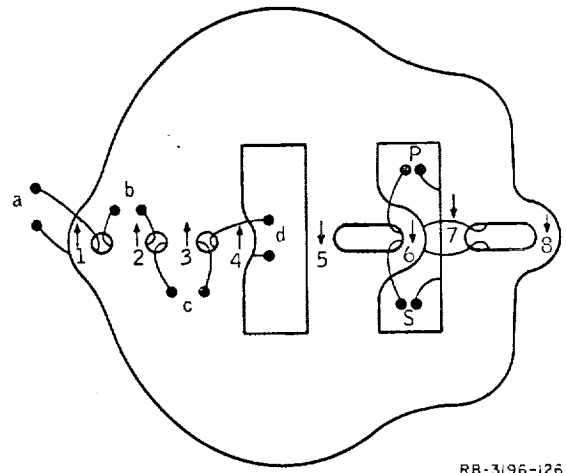
RB-3196-126

FIG. 2.21   MULTIAPERTURED DEVICE FOR
THE PARITY FUNCTION

99

closure path. Similarly, if any two of the inputs are true, Legs 5 and 6 will be switched upward, and so on. Four cases of interest for the flux states in Legs 5, 6, 7, and 8 after the inputs have been applied are listed as Fig. 2.22. Examination of this figure shows that if one input is true, Legs 5 and 6 unblock the aperture they contain; similarly, if three inputs are true, Legs 7 and 8 unblock the rightmost aperture. In all other cases shown in Fig. 2.22, both apertures on the right side of Fig. 2.21 are blocked.

RB-3196-118

FIG. 2.22 FLUX STATES FOR THE
MULTIAPERTURED
DEVICE

As a second phase in the operation, a limited-amplitude pulse P is applied to the primary winding that threads these two apertures. From the foregoing explanation, it should be evident that a secondary signal S is delivered provided that an odd number of inputs were true. Hence, the parity function is produced.

The structure of Fig. 2.21 can be generalized to any even number of inputs, say $2k$, by having $2k - 1$ holes on the left side and k holes on the right side. There are of course some practical limits on the maximum value of k. If a parity function of an odd number of variables is required, one input may be left unused in a structure for the next higher even number of variables.

The drawing of Fig. 2.21 is intended only to illustrate the principles. The dimensions and path lengths have not been optimized.

99

V SUMMARY AND CONCLUSIONS

The logic has been developed for several recoders that recode a dense binary code into a set of outputs that are suitable inputs to a load-sharing access switch. After doing this, several magnetic implementations have been suggested for these recoders.

From the reliability point of view it is evident that if there should be a fault in an address register bit, the memory operates improperly. This is necessary, for by definition the number in the address register is dense—that is, there is no redundancy. However, the recoder-switch structures are safe against some one or more failures in the access-switch drivers, depending on the situation. It has been demonstrated that except for the access-switch drivers, all-magnetic logic can be used between the address register and the memory drive lines; therefore, it seems reasonable to assume high reliability for these recoder-switch structures.

If some redundancy is allowed in the address register, in many cases it is possible to utilize this redundancy to mask failures in the address register drivers. As this may be done using rather conventional techniques it will not be discussed further in this report.

Code-correction circuits of two types have been developed. One of these approaches is inefficient either in the redundancy ratio or in the number of magnetic cores and in the windings per core, depending on the size of the code. The other approach is more efficient; however, it depends on the successful construction of a proposed multiapertured magnetic device. It is evident that more work remains to be done on code-correction circuits.

It seems reasonable to conclude that read-access switches and code-correction circuits can be designed for a random-access magnetic-core memory in such a way that faults can be masked in the memory-access drivers, in the inhibit drivers, and in the sense amplifiers. It appears that a fairly good protection can be obtained with a modest expenditure of equipment.

The next step would seem to be the consideration of an actual paper design of a reliable memory. This would be most effectively done in connection with a reliable computer system.

PART THREE: APPLICATION OF LOGICAL REDUNDANCY FOR FAULT-MASKING IN THE INHIBIT CORE LOGIC SCHEME

By

J. GOLDBERG

I   INTRODUCTION

A.   PURPOSE OF THE STUDY

There is active interest in the use of the Inhibit-Core logic scheme
for the construction of very reliable space-borne computers, because of
the small number of semiconductor devices required, compared to con-
ventional schemes.   In this scheme the small number of semiconductor
devices are concentrated in a small set of identical amplifiers, while
logical operations are performed in magnetic circuits.   Under stringent
conditions, it may be necessary to apply redundancy in order to lengthen
the expected life of a system.   Inasmuch as transistors and diodes are
presently significantly less reliable than the other components of the
scheme, significant improvement will be obtained by giving special pro-
tection against failures of transistors and diodes.

Part Three of this report describes several methods for applying
redundancy at the system level--$i.e.$, by the implementation of various
logical rules for error correction.   Circuit-design methods for applying
redundancy at the component level will not be considered.   Since the
failures of interest are assumed to occur only in transmission channels
($i.e.$, the amplifiers), the problem essentially may be considered as the
design of error-correcting codes and their implementation in a particular
circuit technique.   The Appendix presents an analysis of the theoretical
improvement in reliability given by various single- and double-error-
correcting codes.   An obvious design goal is that the theoretical im-
provement in reliability should not be vitiated by the unreliability of
any special error-correcting circuits, either by increasing the expected
number of catastrophic failures or by decreasing the operating margins
of the system.   An important constraint is that the new logical operations
should decrease the operating speed as little as possible.

Four schemes for applying redundancy will be described.   The first
employs redundant signal channels, with suitable combinational logic
circuits for encoding and decoding (with error correction).   By enlarging
the functions of certain of the existing inhibit-logic circuits, no
change in system timing is required.   The second scheme employs time

103

redundancy in order to reduce equipment redundancy. No extra channels are used; rather, duplicate sets of signals are transmitted through a non-redundant set of channels, but transposed on successive transmissions, so that a given bit of information is sent over different channels. Although added equipment is needed for the duplication and recombination of the information, the equipment may be realized using components of greater reliability than those used in the channel amplifiers. The third and fourth schemes are "hybrid" combinations of the first two, requiring intermediate amounts of time and equipment.

In all schemes, the redundant combinational logic circuits result in an increase of from two to five times in the number of magnetic cores. It is assumed here that the magnetic components do not fail, but the error-correction schemes described actually apply to an entire channel, including cores and amplifier. In the first scheme, the transistor count increases from two to three times for correction of single errors per group of three to nine amplifiers. In the second scheme, no new transistors are needed, but the time per bit cycle increases by a factor of from 50% to 100% for correction of single errors per group of two amplifiers. More precise statements of cost and performance will be given with the scheme descriptions. The four schemes have different proportions of costs in encoder, decoder, and amplifier equipment, and in time delay, allowing the system designer to choose the redundancy best suited to his particular needs.

Section I-B is a brief description of the basic, non-redundant logic scheme.

B. BRIEF DESCRIPTION OF THE INHIBIT-CORE
    LOGIC SCHEME .

Two representations of the original, non-redundant circuits of the particular form of inhibit-core logic which is the subject of this study, are shown in Figs. 3.1(a) and 3.1(b). The heavy vertical lines in Fig. 3.1(a) represent square-loop magnetic cores. The triangular box represents an amplifier, the faults in which are the subject of the protection scheme to be described. In a system, an amplifier may serve a single variable—$i.e.$, it will represent a flip-flop, or it may serve a set of variables, for instance as a register. The first core or set of cores on the left [represented by the block "Register" in Fig. 3.1(b)] store the values of the logical functions of the computer—$i.e.$, the
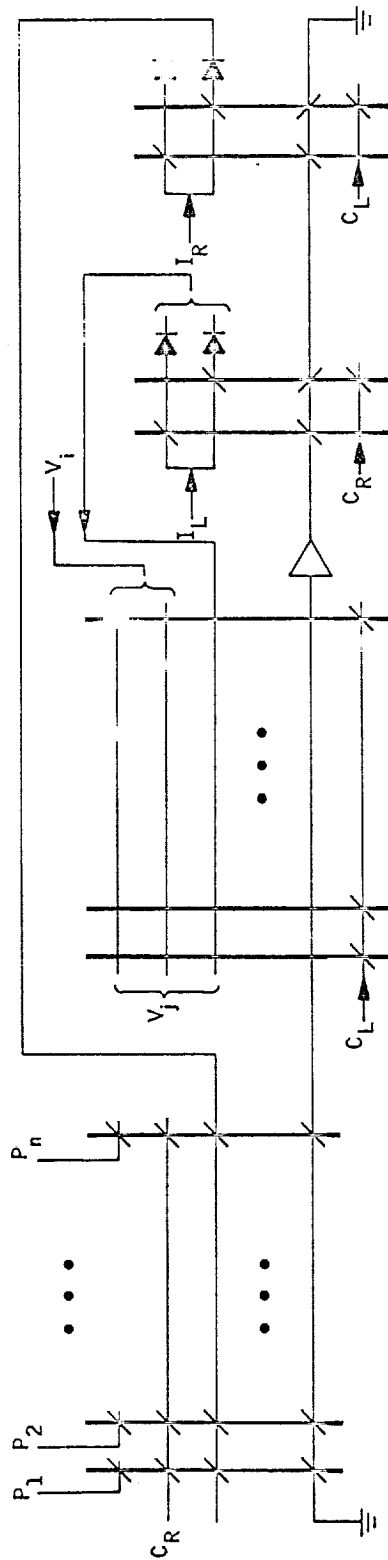
(a) CIRCUIT SCHEMATIC
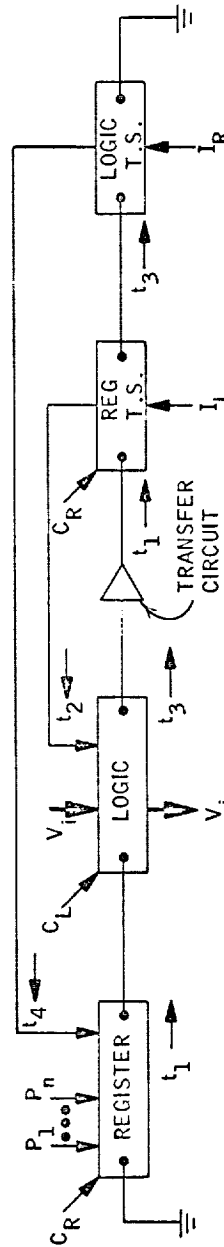
(b) BLOCK AND TIMING DIAGRAM

t_1 READ PRESENT VALUE

t_2 GENERATE NEXT VALUE

t_3 READ NEXT VALUE

t_4 STORE NEXT VALUE

t_1 REG. → REG T.S.

t_2 REG. T.S. → LOGIC

t_3 LOGIC → LOGIC T.S.

t_4 LOGIC T.S. → REG.

FIG. 3.1 BASIC INHIBIT-CORE CHANNEL

RB–3196–98

107

states of a particular flip-flop or register. In this scheme, register bits are scanned sequentially by means of a multiphase clock $(p_1, p_2, \ldots, p_n)$, while single "flip-flop" cores are actuated at every bit time. The second set of cores from the left [represented by the block "Logic" in Fig. 3.1(b)] provide the logic for determining the successive states of the associated function. The function is realized in a disjunctive-AND form, with sufficient redundancy to assure that no more than one core may switch at any time. The input V_i represents the particular subset of the system variables which enter into the logical function. A given system consists of a number of such channels—e.g., thirty, in a system of interest to the client—operating simultaneously. The set of all variables are combined at each bit time to determine the next state of each variable.

In this particular scheme, the same sense amplifier serves both sets of cores, in time-multiplex. Signals from the switching of a register core or a logic core, selected by clocks C_R and C_L respectively, appear on a common sense line, are amplified and detected, and the resulting 0 or 1 signal is stored temporarily in one or the other of two temporary storage blocks [represented as blocks "Reg T.S." and "Logic T.S." in Fig. 3.1(b)]. In addition to temporary storage, these blocks also provide current amplification, so that their outputs may drive the register or logic cores in the inhibit mode. Each bit period is divided into four phases, as follows:

(1) *Read present state*—sense register element, store data in register temporary storage (Reg T.S.) block.

(2) *Determine next state value*—combine Reg T.S. block outputs to set one or none of the logic cores.

(3) *Read next state value*—sense logic cores, store data in logic-temporary storage (Logic T.S.) block.

(4) *Write next state*—copy state of Logic T.S. block into the register element.

C. BRIEF DESCRIPTION OF TWO REDUNDANCY SCHEMES

Part Three of this report discusses some special fault-correcting circuits. In order to give the reader a general picture of how these circuits are used, brief descriptions of several fault-correcting system schemes are given in Sec. I-C. The details of the system organizations are presented in Secs. II-H, III-B, and III-C.

Figure 3.2 represents a redundant system organization of the first kind, based upon a single symmetric-error-correcting code with three independent data bits, (x, y, z), and three parity check digits (a, b, c). With the exception of the Reg T.S. block, the six channels operate as ordinary inhibit-core channels, with no change in timing. The signals produced by the six logic blocks occur simultaneously, at Phase 2, but the values of the a, b, and c signals are designed to be the pair-wise Exclusive-OR functions of x, y, and z. These signals are copied into the individual Logic T.S. blocks, at Phase 3, perhaps with an error due to the failure of one of the amplifiers. When the information is next retrieved (one word time later, for a word register, or one bit time later, for a flip-flop), the six signals are read through the amplifiers and stored in the block labeled "Reg T.S. and Correct." This block combines the data and check digits to accomplish single-error correction, so that if only one amplifier is faulty, either permanently or transiently, the output signals x_c^R, y_c^R, and z_c^R will be the true values of the functions. Other systems will be described in which there are two corrections per bit cycle.

Figure 3.3 is a simplified representation of the second kind of redundancy to be discussed, which will be referred to as the Repetition-Transposition Redundancy Scheme. A scheme for correction of single drop-out-only failures is shown. X and Y are a pair of amplifiers which serve two information sources, A and B. These sources appear duplicated as A_1, A_2, and B_1, B_2, respectively, and the same information is transmitted through X and Y twice, with transposition—i.e., X sends A_1 and B_2, and Y sends B_1 and A_2. The A and B signals are collected separately. Since, in the event of a single amplifier drop-out failure, no more than one of the A receptions and no more than one of the B receptions may drop out, correction consists merely of forming the Inclusive-OR functions of the corresponding received signals. Extension of the scheme to symmetric error correction is immediately obvious—i.e., amplifiers are grouped in triplets, the input data sets are triplicated, and the stored, received sets are corrected by Majority logic. The major feature of this scheme is that there is equipment redundancy only in the magnetic elements, and none in the semiconductor amplifiers, excluding clock drivers.

The third scheme combines features of the first two—i.e., repeated transmissions—with some parity checking.

FIG. 3.2 REDUNDANT SYSTEM ORGANIZATION BASED ON THE (3,3) CODE
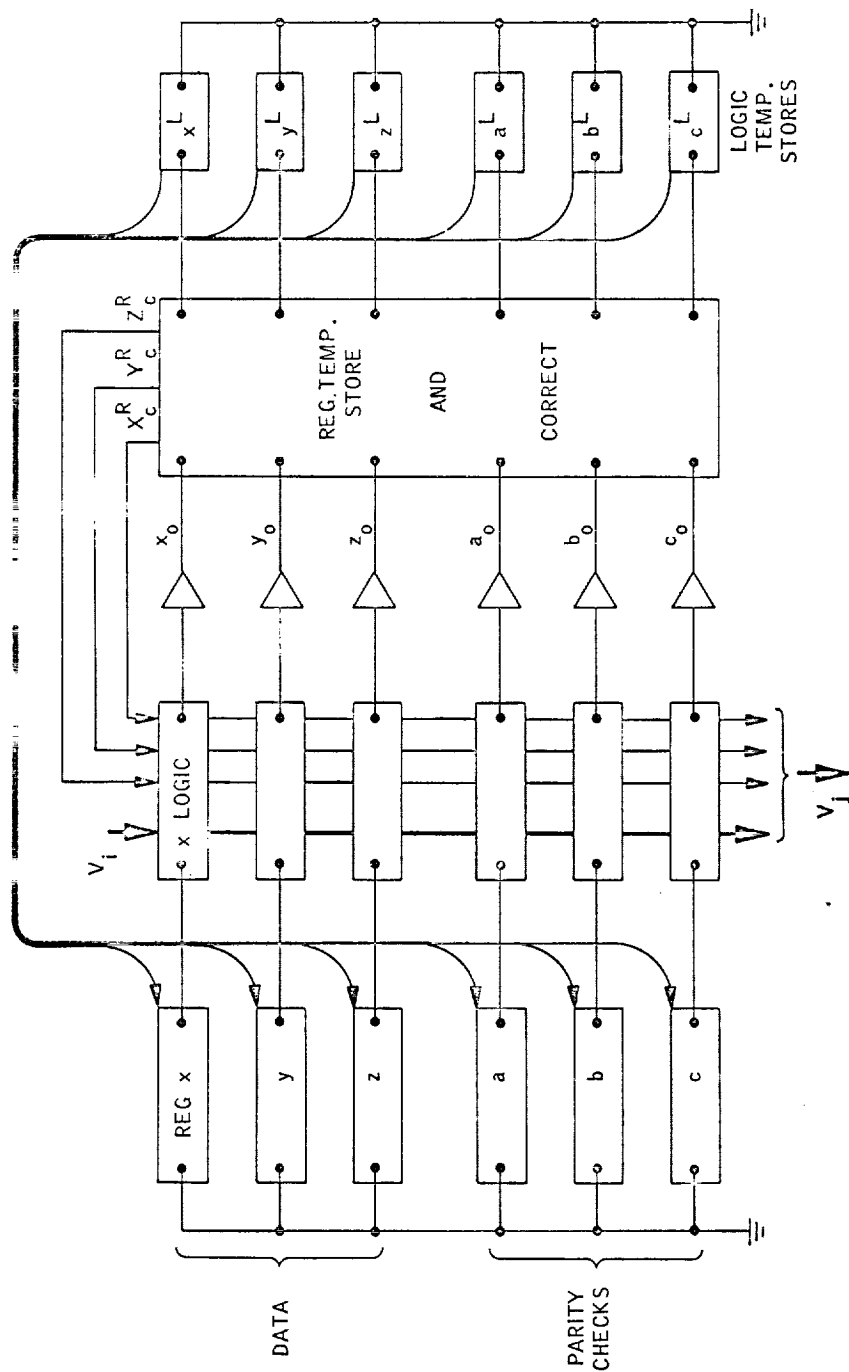
t_1 REG $\rightarrow$ "REG T.S. AND CORRECT"   READ PRESENT VALUES. CORRECT DATA ERRORS

t_2 "REG T.S. AND CORRECT" $\rightarrow$ LOGIC   GENERATE NEXT VALUES

t_3 LOGIC $\rightarrow$ LOGIC T.S.   READ NEXT VALUES

t_4 LOGIC T.S. $\rightarrow$ REG   STORE NEXT VALUES, UNCORRECTED
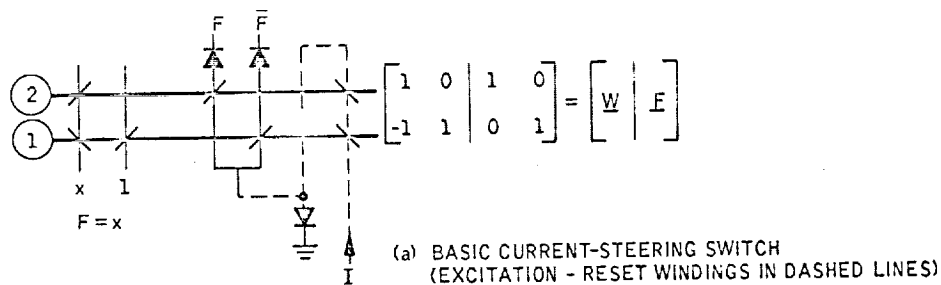
RB-3196-99

110
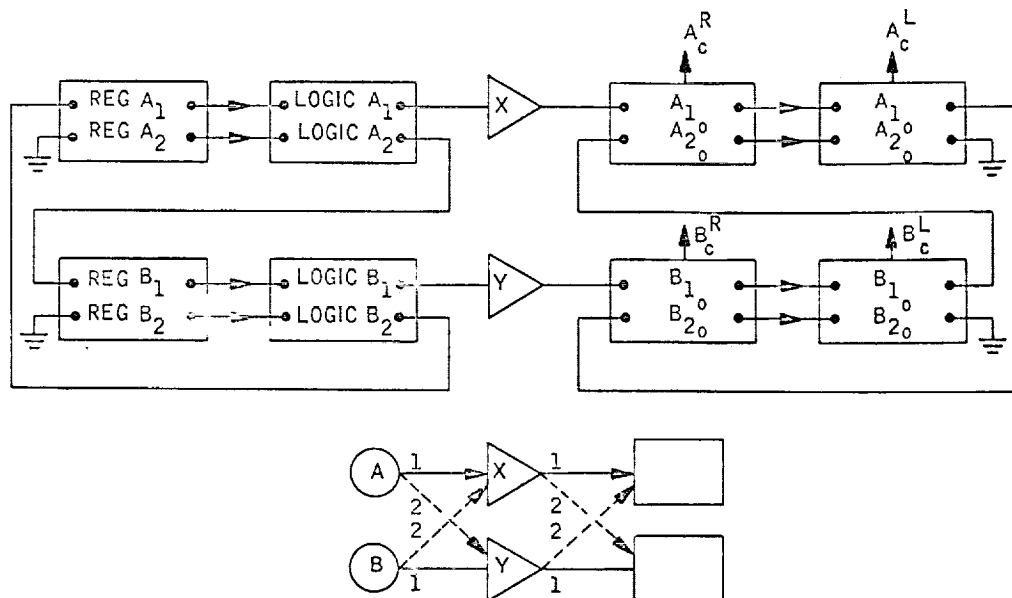
RA-3196-83

FIG. 3.3   SYSTEM ORGANIZATION FOR THE REPETITION-TRANSPOSITION
REDUNDANCY SCHEME

D.   GENERAL REMARKS ON CURRENT-STEERING SWITCHES

The signal at the output of the inhibit-core transfer circuit must
be delayed, and then amplified to a value sufficiently strong to inhibit
the switching of a large number of memory or logic cores.   Both delay
and amplification functions may be realized in a current-steering (C.S.)
switch.   A particular system of interest to JPL uses two-clock-phase
C.S. switches with only these two functions.   Even in the earliest de-
scriptions of C.S. switches,[24] the possibilities were noted of accomplish-
ing logical operations at any or all clock phases.   All the schemes of
this report exploit the opportunity for the logical error-correcting
operations at the input and/or output clock times, so as not to introduce
new logical elements or new time delays.

Figure 3.4 presents a number of simple C.S. switches, using standard
mirror notation, as in Ref. 24.   The horizontal lines represent magnetic
switch cores, the signals x_1, x_2, ... and 1 represent pulsed inputs, and
F and $\overline{F}$ represent the logical function output and its complement.   The
latter carry the current which is steered according to the relative sizes
of the EMF's on the lines linking the cores.   These cores are switched by
the same current, carried by an excitation winding, shown as a dashed line

111

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{F} \end{bmatrix}$$

$F = x$

(a) BASIC CURRENT-STEERING SWITCH
(EXCITATION - RESET WINDINGS IN DASHED LINES)

$$\begin{bmatrix} 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 & 0 \\ -1 & -1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{F} \end{bmatrix}$$

(b) LINEAR-INPUT-LOGIC, DIRECT-OUTPUT CS SWITCH

$F = x_1 \oplus x_2$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 2 \\ 1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{F} \end{bmatrix}$$

(c) DIRECT INPUT, LINEAR-OUTPUT-LOGIC CS SWITCH

$F = \text{MAJORITY}(x_1, x_2, x_3)$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{F} \end{bmatrix}$$

(d) LINEAR-INPUT, LINEAR-OUTPUT-LOGIC CS SWITCH

$F = \text{MAJORITY}(x_3, (x_2 \oplus x_1), x_4)$

RB-3196-89

FIG. 3.4 SYMBOLS, NOTATION, AND LOGICAL OPERATION OF FOUR KINDS OF CURRENT-STEERING SWITCHES

112

in Fig. 3.4(a). Inasmuch as this winding performs no logical function, it will not be shown on the remaining drawings. Beside each diagram is an equivalent winding-matrix representation, in which 1, -1, and 0 represent a positive, a negative, or an absent winding, respectively. The matrix may be partitioned into an input winding matrix $\underline{W}$ and an output winding matrix $\underline{F}$.

It will be assumed that complements of the input variables are not available. In the switches of Figs. 3.4(a) and 3.4(b), the windings are such that one and only one of the cores may be switched by the input signals; thus, although the output lines may link several cores, only one unit of EMF appears on one of the lines when the cores are reset. In Fig. 3.4(b), each input variable drives a separate core, so the number of cores switched may vary from 0 to 3. Considering the bias EMF's produced by Core No. 4, as the number of cores switched range through 0, 1, 2, 3, the EMF on output lines F and $\overline{F}$ range through -1, 0, 1, 2 and 2, 1, 0, -1, respectively. Since the output current is switched onto the relatively more positive line, the F output is excited when two or three—*i.e.*, at least the majority of the three inputs are excited.

In the switch of Fig. 3.4(d), Cores 3 and 1 are switched directly—*i.e.*, independently—by x_3 and x_4, respectively. Core 2a is switched if $x_2 \overline{x}_1 = 1$, and Core 2b if $\overline{x}_2 x_1 = 1$; thus, one of the two will be switched if $x_2 \overline{x}_1 + \overline{x}_2 x_1 = x_2 \oplus x_1 = 1$. At the output Cores 2a and 2b together contribute exactly as does one of the cores in Fig. 3.2(c).

In a paper by Minnick[19] it was shown that a two-level Linear Input circuit could realize any combinational function. The current-steering switches utilized in this discussion permit construction of a 2-level Linear Input circuit in a single structure, using linear input logic for the first level, and linear output logic for the second. This is advantageous because it avoids the cascading of circuits, which would require additional semiconductors and clock phases.

II REDUNDANCY SCHEMES BASED ON
ERROR CORRECTING CODES

A. SOME PERTINENT MATERIAL ON PARITY-CHECK CODES

Figure 3.5 gives three alternative representations of a particular single-symmetric-error-correcting code, in which x, y, and z are three independent data digits, and a, b, and c are redundant checking digits. Figure 3.5(a) is the code table—*i.e.*, the list of all eight standard code symbols. Figure 3.5(b) is the set of logical rules relating the check and the information digits, and Fig. 3.5(c) is an equivalent representation, known as a Parity Check (P.C.) Matrix.

INFORMATION DIGITS			CHECK DIGITS		
x	y	z	a	b	c
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	0	0

$x \oplus y \oplus a = 0$

$y \oplus z \oplus b = 0$

$x \oplus z \oplus c = 0$

$$\begin{array}{cccccc} x & y & z & a & b & c \end{array}$$
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

(a) CODE TABLE   (b) PARITY-CHECK RULES   (c) PARITY-CHECK MATRIX

RB-3196-104

FIG. 3.5 ALTERNATE REPRESENTATIONS OF A CODE

The complexity and thus the cost of the encoding and decoding equipment depends upon the number of *1*'s in the parity check matrix. The most reasonable designs are those whose P.C. matrix contains two *1*'s per column in the data columns (*i.e.*, each data variable appears in only two parity check relations), and from two to four *1*'s per row (*i.e.*, each parity check relation has from two to four input variables).

Conventional error-correction systems are built in two parts, the first part performing checks, and the second executing correction. For

115

the example given, if a single error occurs in the x channel, both its parity check functions $(x_0 \oplus y_0 \oplus a_0)$ and $(x_0 \oplus z_0 \oplus c_0)$ will have the value 1. The rule for changing the observed value x_0, may then be expressed as
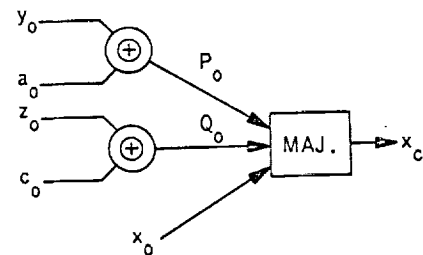
$$x_c = x_0 \oplus [(x_0 \oplus y_0 \oplus a_0)(x_0 \oplus z_0 \oplus c_0)] \quad .$$

The direct logical circuit realization of this rule is shown in Fig. 3.6(a). Note that this calls for three levels of logic. This rule may also be expressed in a form requiring only two levels of logic--$i.e.$,

$$x_c = \text{Majority } [x_0, (y_0 \oplus a_0), (z_0 \oplus c_0)]$$

$$= \text{Majority } (x_0, P_0, Q_0) \text{ [see Fig. 3.6(b)]}$$

$$x_c = x_0 \oplus (x_0 \oplus y_0 \oplus a_0)(x_0 \oplus z_0 \oplus c_0)$$
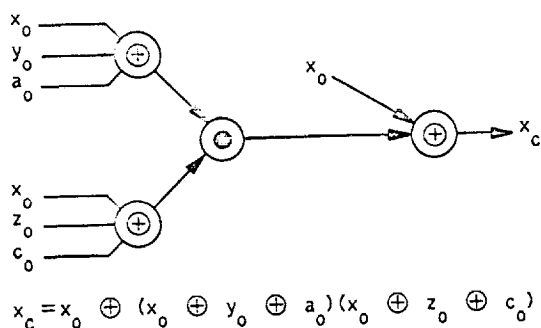
(a) DETECTOR-CORRECTOR SCHEME

$$x_c = \text{MAJ.}(x_0, y_0 \oplus a_0, z_0 \oplus c_0)$$
$$= \text{MAJ.}(x_0, P_0, Q_0)$$

(b) AUGMENTED MAJORITY SCHEME

RB-3196-103

FIG. 3.6   ALTERNATE CORRECTION SCHEMES FOR SINGLE SYMMETRIC ERRORS

which is, fortunately, realizable by current-steering switches employing linear-input logic at input and output. The code-correction schemes based on this rule will be called Augmented Majority Logic Schemes.

In the following sections, correction schemes will be described for single and double symmetric errors and for single asymmetric (drop-outs only) errors within groups of amplifiers.

116

B. DESIGN OF ENCODERS FOR PARITY-CHECK REDUNDANCY

The various parity check redundancy schemes require generation of input signals for the redundant amplifiers which represent the parity function of several of the other inputs. The usual inhibit-core logic circuits do not provide signals of sufficient strength to drive combining logic at the input level. One possibility for realizing the Exclusive-OR function at low signal level is to use a bi-polar amplifier, fed by a special sense wire threading the two core sets in such a way that simultaneous 1 signals cancel. This method requires very careful control of pulse shape and strobing. Because of these circuit difficulties, it is assumed that it is necessary to generate the parity-check functions simultaneously with, and from the same data sources as, the data functions, using the same circuit techniques. Equipment serving this purpose will be called an encoder, by analogy with the communications model.

The number of logic cores required for generating the parity functions depends very much upon the particular functions combined. A study of the functions in a particular design of interest to the client[†] found that most of the variables could be combined in two-input parity checks whose circuit cost was reasonable. Representative results are shown in Table 3.1, which shows the number of logic cores required to produce the individual functions and the two-input parity functions for the indicated combinations.

Table 3.1

NUMBER OF CORES REQUIRED FOR THE GENERATION OF REPRESENTATIVE PARITY FUNCTIONS

FUNCTIONS		NUMBER OF CORES		
X	Y	n_x	n_y	$n_{x \oplus y}$
K_1	K_2	8	6	16
K_3	K_4	13	11	20
K_5	K_6	13	13	24
N	L	5	16	27
E	M	5	5	10
C	P	7	10	34

Only one three-input parity function was tested—*i.e.*, $K_1 \oplus K_2 \oplus K_3$. The number of cores required was 43. This is close to the limit of the present technology, and it is more than twice the number required in the two-input functions containing these variables. Thus the usefulness of three-input parity functions is judged to be marginal.

It is important to note that it is not necessary to apply the same code size and redundancy ratio for all the groups into which a set of

---

[†] Described by the logical equations of D. Rubin.[3]

amplifiers is divided. If the logic of a particular few variables lends itself to three-input parity checks, such possible advantages should be taken, since codes of higher efficiency may then be used. Conversely, some variables may not yield even to two input parity checks. These may be checked by triplication of amplifiers, with, interestingly, no extra costs, since the same logic cores may drive all three amplifiers.

C. CODES AND AUGMENTED MAJORITY DECODERS FOR SINGLE SYMMETRIC ERROR CORRECTION

The codes of this section have parity matrices whose data columns have exactly two 1's, thus their decoders may be realized by "augmented majority" C.S. switches. Three codes of practical importance for groups with 1, 2, and 3 data channels are given in Figs. 3.7, 3.8, and 3.9, respectively.[†] One switch of the kind shown is assumed for each data bit.

(a) PARITY-CHECK MATRIX

(b) LINEAR-OUTPUT-LOGIC CORRECTOR

(c) LINEAR-INPUT-LOGIC CORRECTOR
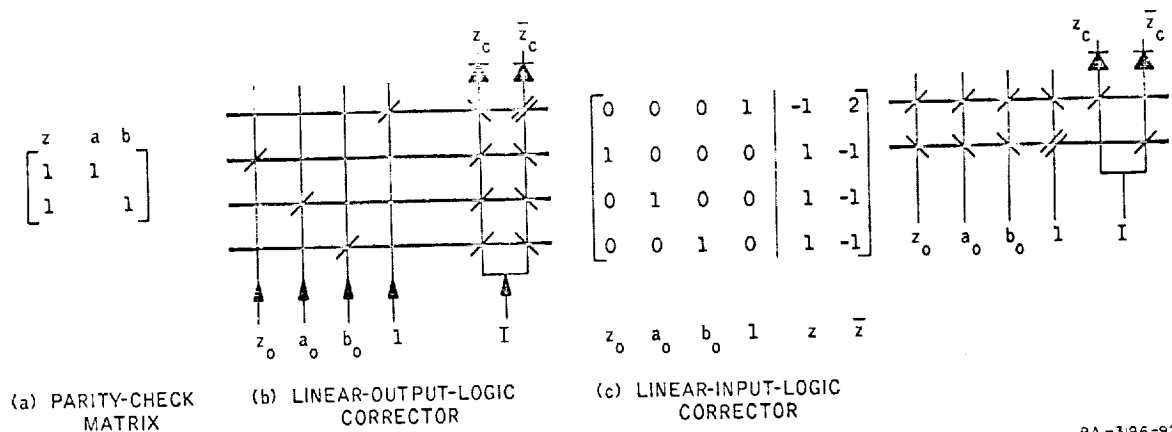
RA-3196-92

FIG. 3.7   SIMPLE MAJORITY LOGIC CORRECTORS

Figure 3.7 describes the familiar triplication-and majority scheme. Figure 3.7(a) is the parity check matrix (z may be considered the data channel), Fig. 3.7(b) is a C.S. correcting switch based on majority logic at the output, and Fig. 3.7(c) is a C.S. correcting switch based on majority logic at the input. The choice between the two is an engineering one.

[†] The codes will be labeled using the notation $(m,k)_i$, where m is the number of data digits, k is the number of check digits, and i is the index for different codes of an m,k class; thus the codes of Figs. 3.7, 3.8, and 3.9 are (1,2), (2,3), and (3,3), respectively.

(a) P MATRIX

(b) WINDING SCHEMATIC (DIRECT-INPUT, LINEAR-OUTPUT LOGIC)

(c) WINDING MATRIX
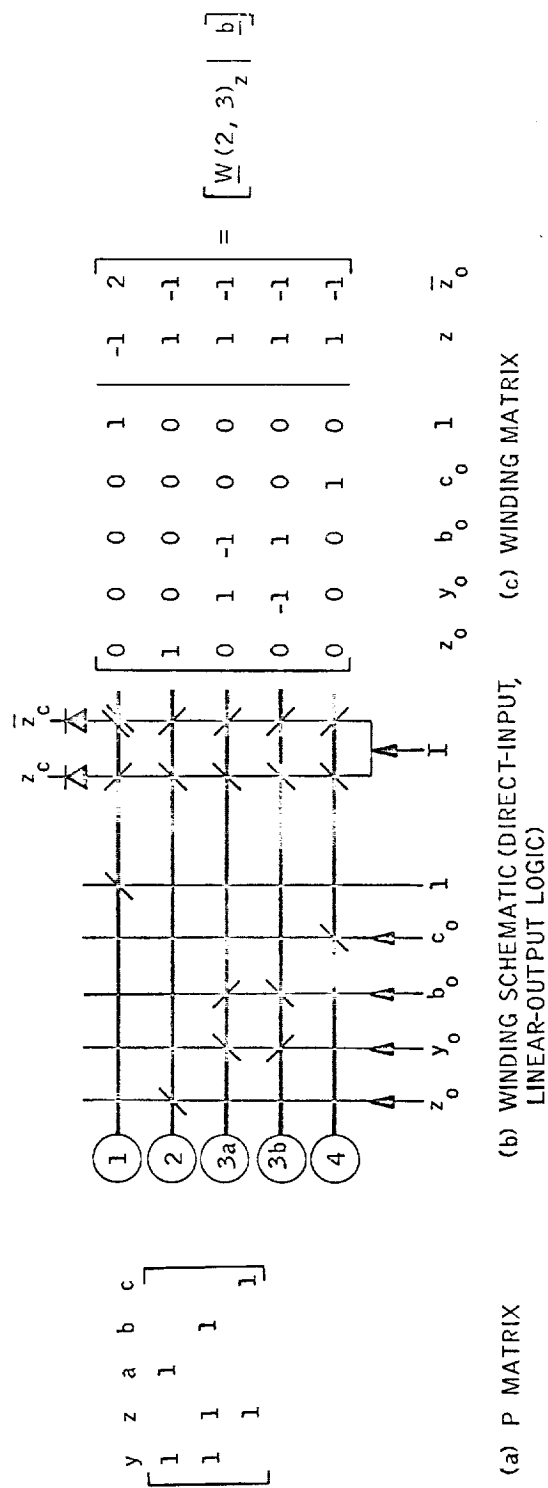
RA-3196-93

FIG. 3.8 AUGMENTED MAJORITY LOGIC CORRECTOR WITH ONE TWO-VARIABLE PARITY FUNCTION

119

(a) P MATRIX

$$\begin{array}{c} x \\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{array} \begin{array}{cccccc} y & z & a & b & c \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & & 1 \end{array}$$

(b) WINDING SCHEMATIC (LINEAR-INPUT, LINEAR-OUTPUT LOGIC)

(c) WINDING MATRIX

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & -1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & -1 \end{bmatrix} = \left[\underline{W}(3,3)_z \; \middle| \; \frac{b}{z} \right]$$

$$\begin{array}{cccccccc} z_0 & y_0 & b_0 & x_0 & c_0 & 1 & z_c & \bar{z}_c \end{array}$$
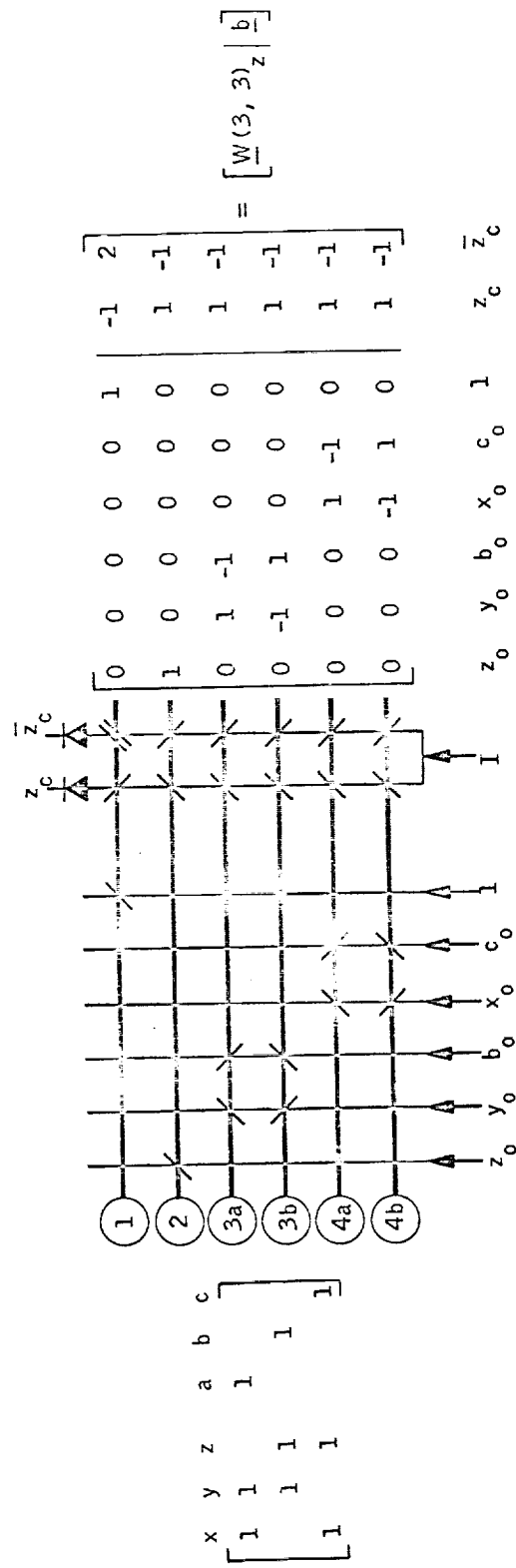
FIG. 3.9   AUGMENTED MAJORITY LOGIC CORRECTOR WITH TWO TWO-VARIABLE PARITY FUNCTIONS

RA-3196-94

120

Figure 3.8 describes a scheme based on a code previously considered by Lofgren.[26] Figure 3.8(a) is the parity matrix, 3.8(b) is the C.S. switch, and 3.8(c) is the switch's winding matrix. Although its amplifier redundancy is higher than that of other codes of this class, it offers the attractive advantage that only one parity function need be generated for every two data variables.

The code of Fig. 3.9 makes more efficient use of the check channels. Its efficiency cannot be exceeded by any code whose parity matrix contains no more than two ones per column and three ones per row.

Parity matrices and winding matrices for correction schemes for codes of 3, 4, and 5 data digits per group are summarized in Fig. 3.10.

Code (3,4), is suitable for the probably rare case in which three data variables have a parity function that is practically realizable. Its advantage is that the inputs for three of the four check-digit channels simply duplicate the data channels. The correction switch requires seven cores while the (3,3) switch requires six. Code (3,4), represents a compromise between the (3,3) and the (3,4) codes. The correction switches for x and z are of the (2,3) kind, and of the (3,3) kind for y.

Table 3.2

EQUIPMENT REDUNDANCY RATIOS FOR CODES WITH $m = 1, 2, 3, 4, 5$

CODE BITS		REDUNDANCY RATIOS		
Info m	Check k	Amplifiers r_A	Encoder r_E	Decoder r_D
1	0	1	1	1
1	2	3	1	2
2	3	2.5	2	2.5
3	3	2	3	3
3	4 (1)	2.33	2	3.5
3	4 (2)	2.33	2.33	2.67
4	4	2	3	3
4	5	2.25	2.5	3.1
5	4	1.8	3	4.5
5	5	2	3	3
5	6	2.2	2.6	2.8

The equipment redundancy ratios of the above schemes are listed in Table 3.2. The ratios for amplifiers, encoder logic, and decoder switches are listed separately. The encoder costs are estimates, assuming a two input parity check encoder has twice the average number of cores, and a three input parity check encoder has three times the average number. The choice of code depends partly upon the relative weights of the three equipments.

Aside from the (5,4) code, no code has a lower amplifier redundancy ratio than the (3,3) code, but several codes of larger group size have the

(a) (3, 4) CODES

$$\underline{P}(3,4)_1 = \begin{array}{ccccccc} x & y & z & a & b & c & d \end{array}$$
$$\begin{bmatrix} 1 & & & 1 & & & \\ & 1 & & & 1 & & \\ & & 1 & & & 1 & \\ 1 & 1 & 1 & & & & 1 \end{bmatrix}$$

$$[\underline{W}(3,4)_z \mid \underline{b}] = \begin{array}{cccccc} z_o & x_o & y_o & d_o & c_o & b \end{array}$$
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -2 \\ 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \underline{W}_x, \underline{W}_y \text{ SIMILAR}$$

$$\underline{P}(3,4)_2 = \begin{array}{ccccccc} x & y & z & a & b & c & d \end{array}$$
$$\begin{bmatrix} 1 & & & 1 & & & \\ 1 & 1 & & & 1 & & \\ & 1 & 1 & & & 1 & \\ & & 1 & & & & 1 \end{bmatrix}$$

$\underline{W}_z, \underline{W}_x$ SAME FORM AS $\underline{W}(2,3)_z$

$\underline{W}_y$ SAME FORM AS $\underline{W}(3,3)_z$

(b) (4, 4) CODE

$$\underline{P}(4,4) = \begin{array}{cccccccc} w & x & y & z & a & b & c & d \end{array}$$
$$\begin{bmatrix} 1 & 1 & & & 1 & & & \\ & 1 & 1 & & & 1 & & \\ & & 1 & 1 & & & 1 & \\ 1 & & & 1 & & & & 1 \end{bmatrix} \quad \underline{W}_w, \underline{W}_x, \underline{W}_y, \underline{W}_z \text{ SAME FORM AS } \underline{W}(3,3)_z$$

(c) (4, 5) CODES

$$\underline{P}_1(4,5) = \begin{array}{ccccccccc} w & x & y & z & a & b & c & d & e \end{array}$$
$$\begin{bmatrix} 1 & & & & 1 & & & & \\ 1 & 1 & & & & 1 & & & \\ & 1 & 1 & & & & 1 & & \\ & & 1 & 1 & & & & 1 & \\ & & & 1 & & & & & 1 \end{bmatrix}$$

$\underline{W}_w, \underline{W}_z$ SAME FORM AS $\underline{W}(2,3)_z$

$\underline{W}_x, \underline{W}_y$ SAME FORM AS $\underline{W}(3,3)_z$

$$P_2(4,5) = \begin{array}{ccccccccc} w & x & y & z & a & b & c & d & e \end{array}$$
$$\begin{bmatrix} 1 & & & & 1 & & & & \\ & 1 & & & & 1 & & & \\ & & 1 & & & & 1 & & \\ 1 & & & 1 & & & & 1 & \\ & 1 & 1 & 1 & & & & & 1 \end{bmatrix}$$

$$[\underline{W}_2(4,5)_z \mid \underline{b}] = \begin{array}{ccccccc} z_o & w_o & d_o & x_o & y_o & e_o & 1 \end{array}$$
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & -2 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \end{bmatrix}$$

$\underline{W}_w$ SAME FORM AS $\underline{W}(2,3)_z$

$\underline{W}_x, \underline{W}_y$ SAME FORM AS $\underline{W}_2(3,3)_z$

FIG. 3.10   PARITY-CHECK AND WINDING MATRICES FOR ERROR-CORRECTING CODES OF 1, 2, 3, 4, AND 5 INFORMATION BITS

119

(d) (5, 4) CODE

$$
\begin{array}{ccccccccc}
v & w & x & y & z & a & b & c & d \\
\end{array}
$$

$$
\begin{bmatrix}
1 & 1 & 1 & & & 1 & & & \\
& & 1 & 1 & 1 & & 1 & & \\
1 & & & 1 & & & & 1 & \\
& 1 & & & 1 & & & & 1
\end{bmatrix}
$$

$\underline{W}_v,\ \underline{W}_w,\ \underline{W}_y,\ \underline{W}_z,\ \text{SEE}\ \underline{W}_2(4,5)_z$

$$
\left[\underline{W}(5,4)_x \mid b\right] =
\begin{array}{ccccccccc}
x_o & v_o & w_o & a_o & y_o & z_o & b_o & \text{BIAS} \\
\end{array}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & -2 \\
0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & -2 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 1 & 0
\end{bmatrix}
$$

(e) (5, 5) CODE

$$
\begin{array}{cccccccccc}
v & w & x & y & z & a & b & c & d & e \\
\end{array}
$$

$$
\begin{bmatrix}
1 & 1 & & & & 1 & & & & \\
& 1 & 1 & & & & 1 & & & \\
& & 1 & 1 & & & & 1 & & \\
& & & 1 & 1 & & & & 1 & \\
1 & & & & 1 & & & & & 1
\end{bmatrix}
$$

$\underline{W}_v,\ \underline{W}_w,\ \underline{W}_x,\ \underline{W}_y,\ \underline{W}_z,\ \text{SEE}\ \underline{W}(3,3)_z$

(f) (5, 6) CODE

$$
\begin{array}{ccccccccccc}
v & w & x & y & z & a & b & c & d & e & f \\
\end{array}
$$

$$
\begin{bmatrix}
1 & & & & & 1 & & & & & \\
1 & 1 & & & & & 1 & & & & \\
& 1 & 1 & & & & & 1 & & & \\
& & 1 & 1 & & & & & 1 & & \\
& & & 1 & 1 & & & & & 1 & \\
& & & & 1 & & & & & & 1
\end{bmatrix}
$$

$\underline{W}_v,\ \underline{W}_z,\ \text{SEE}\ \underline{W}(2,3)_z$

$\underline{W}_w,\ \underline{W}_x,\ \underline{W}_y,\ \text{SEE}\ \underline{W}(3,3)_z$

FIG. 3.10   CONCLUDED

same redundancy. It is natural to ask whether smaller or larger group sizes are to be preferred, given the same redundancy and assuming single-error correction within a group. The preference should be given to the smaller group, because more double errors may be corrected the greater the number of groups into which a given number of data channels are divided.

This may be illustrated by comparing the (6,6) and (3,3) codes (Fig. 3.11). As shown, nine double errors may be corrected in the (6,6) code, while the same six data channels and six check channels, in two groups of three-and-three, allow for thirty-six double errors.

123

120

(a) UNIFIED 6-BIT CODE GROUP

$$\begin{array}{cccccccccccc} u & v & w & x & y & z & & a & b & c & d & e & f \\ \end{array}$$

$$\begin{bmatrix} 1 & 1 & & & & & & 1 & & & & & \\ & 1 & 1 & & & & & & 1 & & & & \\ & & 1 & 1 & & & & & & 1 & & & \\ & & & 1 & 1 & & & & & & 1 & & \\ & & & & 1 & 1 & & & & & & 1 & \\ 1 & & & & & 1 & & & & & & & 1 \end{bmatrix}$$

CORRECTABLE DOUBLE-ERROR SETS (WITH NO SPURIOUS EFFECTS ON OTHER VARIABLES)

u, x   v, y   w, z   x, a   z, c

u, c   v, d   w, e   y, b

NOTE: u, w CAUSES FALSE CHANGE IN v, ETC.

(b) ALTERNATIVE TWO THREE-BIT GROUPS

$$\begin{array}{cccccc} u & v & w & a & b & c \\ \end{array}$$

$$\begin{bmatrix} 1 & 1 & & 1 & & \\ & 1 & 1 & & 1 & \\ 1 & & 1 & & & 1 \end{bmatrix}$$

CORRECTABLE DOUBLE-ERROR SETS

ANY ONE OF (u, v, w, a, b, c), AND

ANY ONE OF (x, y, z, d, e, f)

(36 SETS)

$$\begin{array}{cccccc} x & y & z & d & e & f \\ \end{array}$$

$$\begin{bmatrix} 1 & 1 & & 1 & & \\ & 1 & 1 & & 1 & \\ 1 & & 1 & & & 1 \end{bmatrix}$$

RB-3196-97

FIG. 3.11   COMPARISON OF DOUBLE-ERROR-CORRECTION CAPABILITIES FOR ONE SIX-BIT GROUP VERSUS TWO THREE-BIT GROUPS

D.   SINGLE-ERROR-CORRECTING SWITCHES FOR CODES WITH PARITY-CHECK MATRICES HAVING THREE 1'S PER COLUMN

Although increasing the number of parity check relations into which the data channels may enter makes possible the more efficient use of re-dundant channels, the correction switches become considerably more com-plicated. An important illustration is given by the (4,3) Hamming Code, whose parity matrix (with 0's deleted for readability) may be written as:

$$\begin{array}{ccccccc} w & x & y & z & a & b & c \\ \end{array}$$

$$\begin{bmatrix} 1 & 1 & & 1 & 1 & & \\ 1 & & 1 & 1 & & 1 & \\ & 1 & 1 & 1 & & & 1 \end{bmatrix} \; .$$

124

Here, correction requires inspection of three parity relations, rather than two as in the previously considered codes. Note, for example, that although w appears explicitly in only two relations, the presence of z in both of those relations requires a third check. Thus, the rule for correction of w is:

$$w_c = w_0 \oplus (w_0 \oplus P_{w_0})(w_0 \oplus Q_{w_0})\overline{R}_{w_0}$$

where

$$P_{w_0} = x_0 \oplus z_0 \oplus a_0$$

$$Q_{w_0} = y_0 \oplus z_0 \oplus b_0$$

$$R_{w_0} = y_0 \oplus z_0 \oplus c_0 \quad .$$

There are two decompositions of this expression which might usefully exploit magnetic core circuitry. One is a two-level linear-input logic circuit with the following structure:[†]

$$[1, R, 2 (P, Q, R * 1) ** P, Q, 3w] \quad ,$$

in which the terms on the right-hand side of the * or ** represent the setting inputs, and those on the left the inhibiting inputs, each group weighted, summed, and tested for "equal-to-or-greater than 1." A circuit implementing this logic is shown in Fig. 3.12. Notice that, for simplicity of illustration, the input windings for x, y, z, a, b, and c are not shown, and the sets of cores required for generating the intermediate R, P, and Q variables are represented as single cores. The circuit is not only complex, but it requires an additional diode and an additional clock period, so it is probably of little practical interest.

Another decomposition of the above expression for w_c is

$$w_c = R_{w_0} w_0 + \overline{R}_{w_0} \text{Maj} (w_0, P_{w_0}, Q_{w_0}) \quad .$$

[†] Derived from entry 156, Table II, Ref. 19.

RA-3196-85
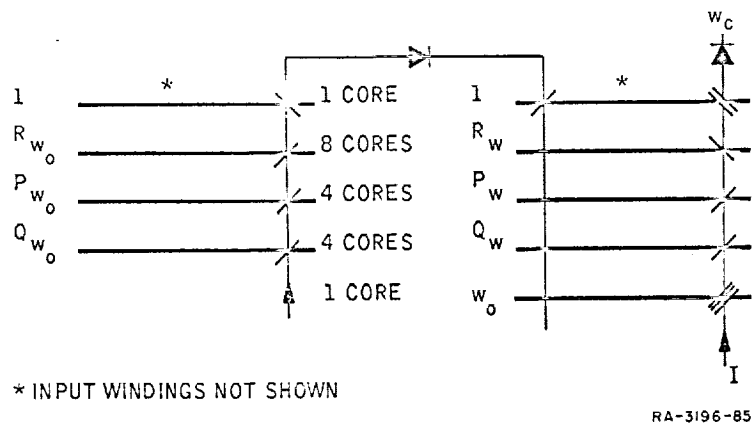
FIG. 3.12 TWO-LEVEL CORRECTION SWITCH FOR THREE-ONES-
PER-COLUMN-P MATRIX CODES

This may be realized by a hybrid circuit, using linear input logic
for the majority function, and diode logic for the OR function, as in
Fig. 3.13. Figure 3.13(a) shows a switch suitable for correcting the
w, x, or y information, and the switch of 3.13(b) is for correcting z.
The same abbreviations are used as in Fig. 3.12, but Fig. 3.13(c) illus-
trates a typical winding matrix for the P_{w_0} and R_{w_0} variables. Extra
diodes are used, but no extra clock period. Nevertheless, the size of
the switch may be beyond practical realization.

E.   ERROR-CORRECTING SWITCHES BASED
      ON COMPLETE DECODING[†]

All of the switches discussed in the previous sections were based
on the parity check relations among the code digits. An alternative
approach is to exploit the distance-sensitive properties of Linear Input
logic to isolate individual code "letters," as follows. Let a separate
Linear Input element be assigned to each code letter, and let the inputs
be weighted +1 for digits which appear as 1 in the letter and -1 for
digits which appear as 0 in the letter. Adjust the threshold t so that
$n_1 - n_0 - t = (d + 1)/2$, where n_1 is the number of 1's in the letter, n_0 the
number of 0's and d is the code distance. The effect of either a false
1 or a false 0 is to diminish the drive by one unit. So long as the
number of digits changed is less than $(d - 1)/2$, the element will still
be selected.

[†] See also Sec. IV-B, Part Two of this report.

(a) w, x, y – TYPE SWITCH

(b) SWITCH FOR Z-CHANNEL
CORRECTION

* INPUT WINDINGS NOT SHOWN

(c) ILLUSTRATIONS OF 3-INPUT (P^w) AND
4-INPUT R^w PARITY-CHECK SET WINDINGS

RB-3196-84
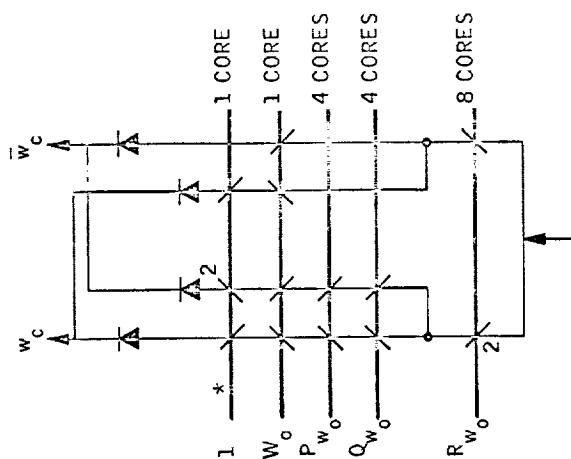
FIG. 3.13   AUGMENTED-MAJORITY CORRECTION SWITCH FOR THREE-ONES-PER-COLUMN-P
MATRIX CODES [Shown for the (4,3) Hamming Code]

The switches resulting from this approach are attractive for small code groups, but since the number of elements needed for an m bit code is 2^m, the utility of the approach quickly vanishes for increasing m. Figure 3.14 illustrates distance-3 switch designs, for the (2,3), (3,3), and (4,3) codes. One possible advantage of this realization is that several output lines may share the same set of cores, rather than having to use separate code correction switches as in previous sections. The electrical design of such a shared-output switch is somewhat more difficult, but it appears reasonable. The (2,3) switch requires fewer total cores and windings than the "augmented majority" version, but it requires more inputs to its linear-input section, increasing the design difficulty somewhat. The (3,3) switch requires fewer cores than the total of the three switches of its previously stated counterpart, but it is a larger switch, thus it is more difficult to design. Both versions of the above two cases should be considered by the circuit designer. The (4,3) switch shown in Fig. 3.14 is superior to the parity-based version, in the number of cores and diodes, but it is still very large.

F. DOUBLE-ERROR-CORRECTING SWITCHES

Switch designs for distance 5 codes based on the parity-check relations are very complex and appear impractical. For single-data-bit codes, a majority type switch is feasible of the input-logic or the output-logic type. Both versions are shown in Fig. 3.15. For two-data-bit codes, a switch based on the complete decoding of the four valid code letters, as in Fig. 3.16, also appears feasible.

G. ASYMMETRIC-ERROR CORRECTING SWITCHES

If an amplifier may be trusted to fail only in one direction, the cost of error correction may be reduced. Each channel is duplicated; in the "drop-outs only" case, the correction switch simply produces the OR function of the duplicate channels, while in the "false-ones only" case, the correction switch produces the AND function. The first type is illustrated in Fig. 3.17. The input encoding logic is the same for both channels, and might be a single circuit. There is no advantage in using groups larger than two.

The simplicity and low cost of this correction scheme suggests that it would be profitable to design the amplifier so that one kind of error
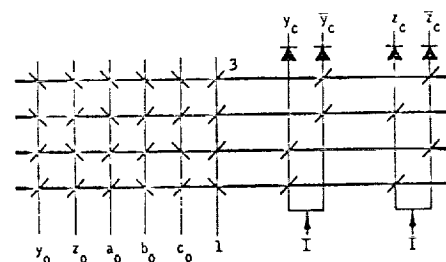
(a) m = 2 CODE

P MATRIX

y	z	a	b	c
1		1		
1	1		1	
	1			1

CODE SET

y	z	a	b	c
0	0	0	0	0
0	1	0	1	1
1	0	1	1	0
1	1	1	0	1

LINEAR-INPUT, DIRECT-OUTPUT-CORRECTING SWITCH

(b) m = 3, k = 3 CODE

$\underline{P} =$

x	y	z	a	b	c
1	1		1		
	1	1		1	
1		1			1

$\underline{C} =$

x	y	z	a	b	c
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	0	0

WINDING MATRIX

x_0	y_0	z_0	a_0	b_0	c_0	BIAS	x_c	$\bar{x}_c$	y_c	$\bar{y}_c$	z_c	$\bar{z}_c$
-1	-1	-1	-1	-1	-1	2	0	1	0	1	0	1
-1	-1	1	-1	1	1	-1	0	1	0	1	1	0
-1	1	-1	1	1	-1	-1	0	1	1	0	0	1
-1	1	1	1	-1	1	-2	0	1	1	0	1	0
1	-1	-1	1	-1	1	-1	1	0	0	1	0	1
1	-1	1	1	1	-1	-2	1	0	0	1	1	0
1	1	-1	-1	1	1	-2	1	0	1	0	0	1
1	1	1	-1	-1	-1	-1	1	0	1	0	1	0

(c) m = 4, k = 3 (HAMMING) CODE

P MATRIX

w	x	y	z	a	b	c
1	1	1		1		
1	1		1		1	
	1	1	1			1

WINDING MATRIX

w	x	y	z	a	b	c	BIAS	w_c	$\bar{w}_c$	x_c	$\bar{x}_c$	y_c	$\bar{y}_c$	z_c	$\bar{z}_c$
-1	-1	-1	-1	-1	-1	-1	2	0	1	0	1	0	1	0	1
-1	-1	-1	1	-1	1	1	-1	0	1	0	1	0	1	1	0
-1	-1	1	-1	1	-1	1	-1	0	1	0	1	1	0	0	1
-1	-1	1	1	1	1	-1	-2	0	1	0	1	1	0	1	0
-1	1	-1	-1	1	1	1	-2	0	1	1	0	0	1	0	1
-1	1	-1	1	1	-1	-1	-1	0	1	1	0	0	1	1	0
-1	1	1	-1	-1	1	-1	-1	0	1	1	0	1	0	0	1
-1	1	1	1	-1	-1	1	-2	0	1	1	0	1	0	1	0
1	-1	-1	-1	1	1	-1	-1	1	0	0	1	0	1	0	1
1	-1	-1	1	1	-1	1	-2	1	0	0	1	0	1	1	0
1	-1	1	-1	-1	1	1	-2	1	0	0	1	1	0	0	1
1	-1	1	1	-1	-1	-1	-1	1	0	0	1	1	0	1	0
1	1	-1	-1	-1	-1	1	-1	1	0	1	0	0	1	0	1
1	1	-1	1	-1	1	-1	-2	1	0	1	0	0	1	1	0
1	1	1	-1	1	-1	-1	-2	1	0	1	0	1	0	0	1
1	1	1	1	1	1	1	-5	1	0	1	0	1	0	1	0

RC-3196-97

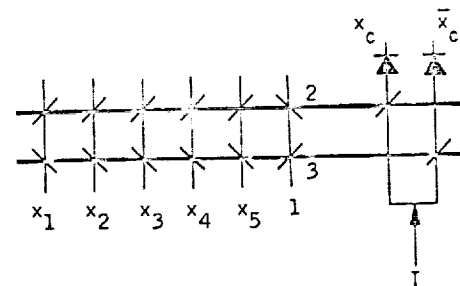FIG. 3.14 SINGLE-SYMMETRIC-ERROR-CORRECTING SCHEMES
BASED ON COMPLETE DECODING
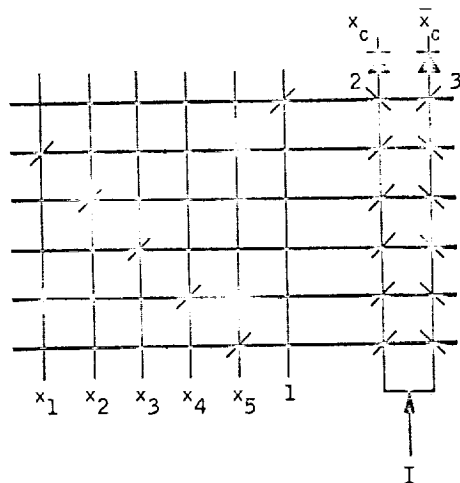
(a) LINEAR OUTPUT LOGIC

(b) LINEAR INPUT LOGIC

RB-3:96-100

FIG. 3.15   THREE-OUT-OF-FIVE MAJORITY DOUBLE-ERROR CORRECTION SCHEME

x	y	a	b	c	d	e	f
1		1					
1			1				
1	1			1			
1	1				1		
	1					1	
	1						1

P MATRIX

x	y	a	b	c	d	e	f
0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1
1	0	1	1	1	1	0	0
1	1	1	1	0	0	1	1

CODE SET

x	y	a	b	c	d	e	f	BIAS	x_c	$\bar{x}_c$	y_c	$\bar{y}_c$
-1	-1	-1	-1	-1	-1	-1	-1	3	0	1	0	1
-1	1	-1	-1	1	1	1	1	-2	0	1	1	0
1	-1	1	1	1	1	-1	-1	-2	1	0	0	1
1	1	1	1	-1	-1	1	1	-3	1	0	1	0

WINDING MATRIX

RB-3:96-101

FIG. 3.16   TWO-INFORMATION-CHANNEL DOUBLE-ERROR-CORRECTION SCHEME
BASED ON COMPLETE DECODING

130

(a) LINEAR OUTPUT LOGIC

(b) LINEAR INPUT LOGIC

(c) SYSTEM ORGANIZATION

RA-3196-87
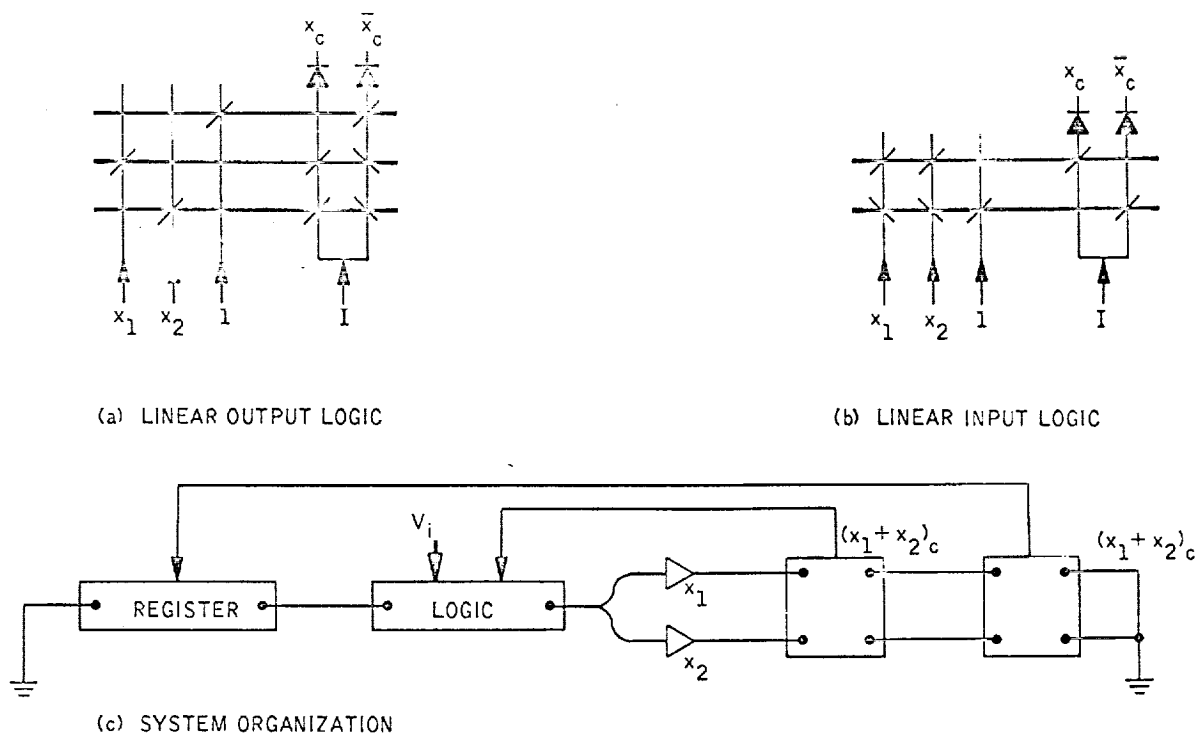
FIG. 3.17   CORRECTION SCHEMES FOR SINGLE-DROP-OUT ERRORS

is much less likely than the other kind—*e.g.*, by using fuses to convert short-circuits to open circuits.[†]

H.   SYSTEM ORGANIZATION FOR CODE-TYPE ERROR CORRECTION

In the inhibit-core scheme to which these methods are addressed, the amplifiers are used twice per bit cycle, once to read the present state of the variables from registers, and then to read the next-state values, obtained from the logic circuits. If it may be assumed that errors will not occur in different amplifiers on the two successive phases of a bit cycle, the information need be corrected only once—*i.e.*, when the output of the temporary register information is used to form the next-state values in logic. Nevertheless there may be some equipment advantage in making two corrections per cycle. In the single-correction

---

[†] See Sec. V-B, Part One.

system, extra registers are needed to hold the parity information for the subsequent correction operation, but the logic temporary-storage switches are simple. In the two-corrections system, only data registers are needed, but the logic T.S. switches must be correctors.

These considerations are illustrated in the simple Majority Redundancy scheme, shown in Fig. 3.18. The lines labeled x_c^R and x_c^L represent the corrected register and logic variables, and V_i and V_j
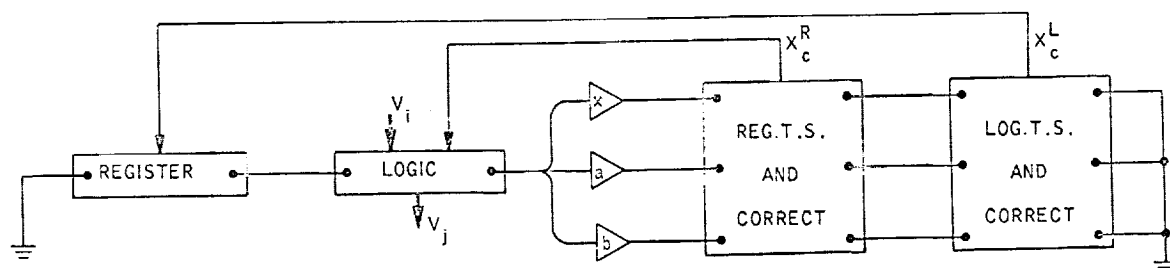
(a) 2 CORRECTIONS PER CYCLE
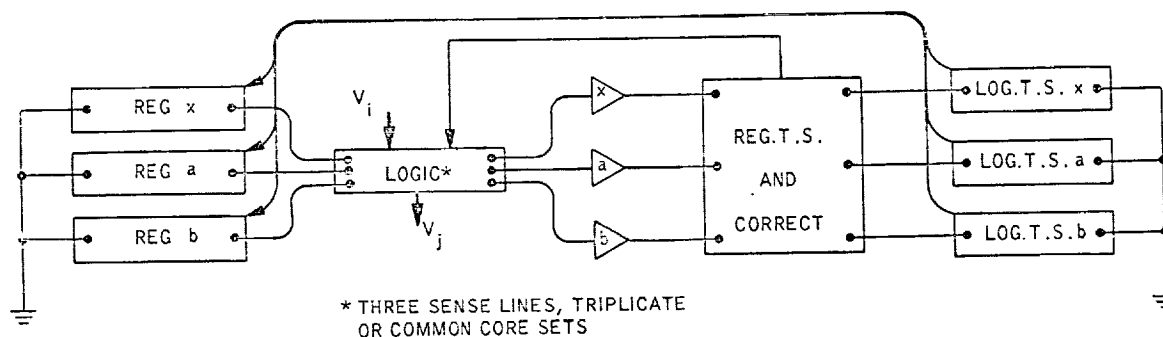
(b) 1 CORRECTION PER CYCLE (AT LOGIC INHIBIT TIME)

* THREE SENSE LINES, TRIPLICATE
OR COMMON CORE SETS

RB-3196-91

FIG. 3.18 ALTERNATE SYSTEM ORGANIZATIONS FOR SIMPLE MAJORITY REDUNDANCY

represent the subsets of the other system variables—with and without x—that determine the successive values of x. The rest of the notation is that given in Sec. I, in connection with Fig. 3.2. The choice between the two alternatives of one or two corrections per cycle is based upon the relative equipment and design costs of two registers and three single temporary storage circuits versus one correcting-type T.S. circuit. In

the single-correction scheme, the logic circuits may logically be shared by the three sense lines.

Figure 3.19 illustrates the two system choices for the (2,3) code. The block labeled "Reg T.S. and Correct" consists of error-correcting switches of the kinds described in the previous section, either one, using "complete decoding," or two, using "augmented majority." A new

(a) TWO CORRECTIONS PER CYCLE

* 2 SENSE LINES, DUPLICATE OR
  COMMON CORE SETS

(b) ONE CORRECTION PER CYCLE
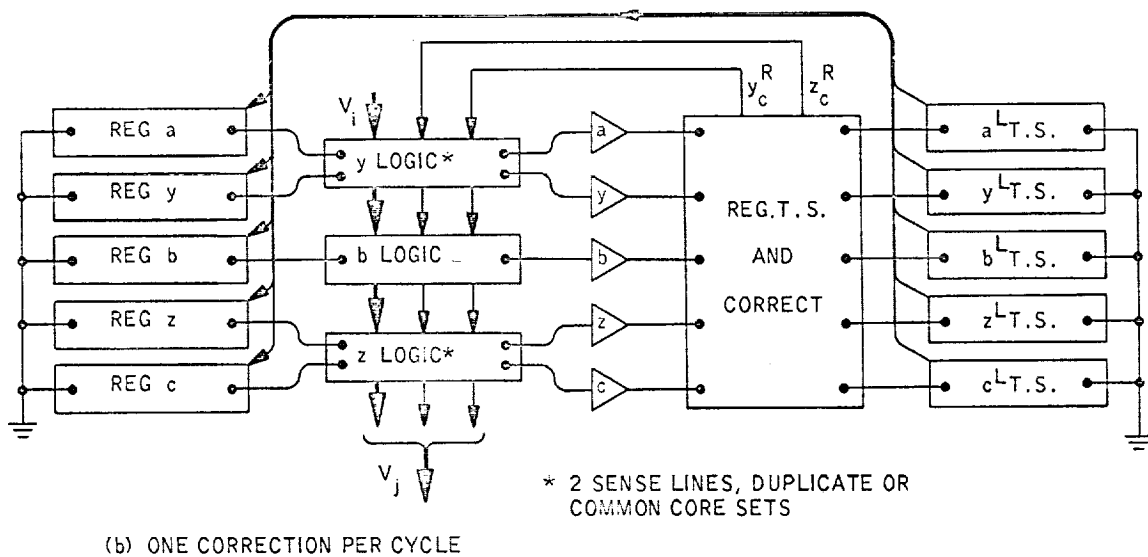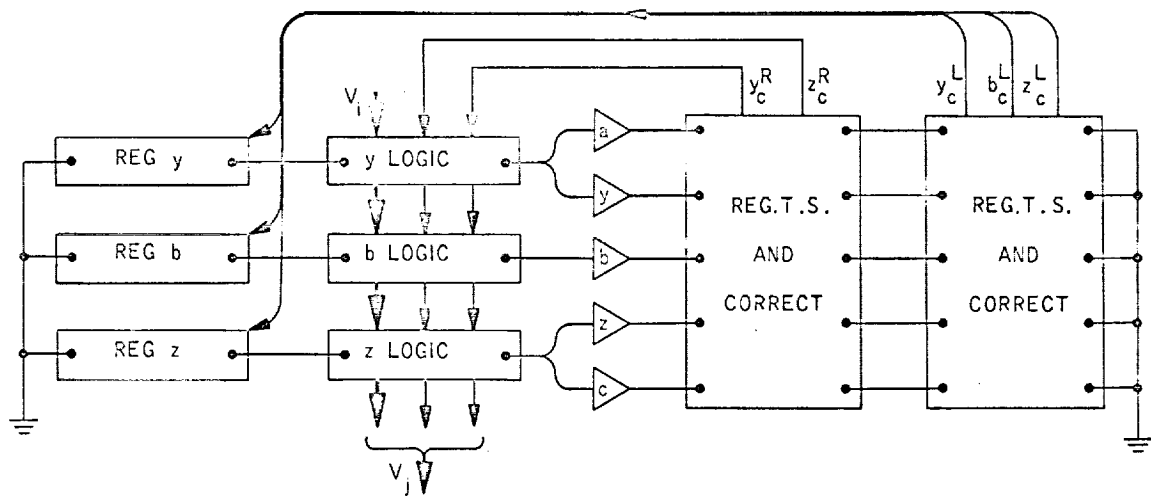
RB-3196-88

FIG. 3.19   ALTERNATE SYSTEM ORGANIZATIONS FOR THE (2,3) CODE

133

problem appears in the requirement for generating a corrected parity digit signal in the two-corrections-per-cycle scheme. In the (2,3) code, the only unique parity channel is b. Its correction rule is more complex than that for the data channels—i.e.,

$$b_c = b_0 \oplus (y_0 \oplus z_0 \oplus b_0)(y_0 \oplus \overline{a_0})(z_0 \oplus \overline{c_0}) \quad .$$

The most economical realization of this function is

$$b_c = (1, y_0, a_0 * z_0, b_0, c_0) + (1, z_0, c_0 * y_0, a_0, b_0) \quad .$$

This requires only two cores, with the linear logic functions at their inputs and the OR logic at their outputs.[†] A hybrid system is also admissible, in which the b channel would be corrected once per cycle, and the others, twice.

[†] This form was arrived at by the "complete decoding" approach. "Straightforward" simplification of the initial formula yields the form
$$b_c = b_0(y \oplus a + y \oplus z + z \oplus c) + y\bar{z}a\bar{c} + \bar{y}z\bar{a}c \quad ,$$
which has a much more expensive realization.

III ERROR CORRECTION BY REPETITION AND TRANSPOSITION

A. GENERAL DESCRIPTION

The schemes of the previous section employed separate channels to
carry redundant information for correction of errors in the data channels.
In order to minimize the number of extra channels, group codes were em-
ployed, which required special logic circuitry for encoding and decoding.
Codes were found that allowed the accomplishing of the logical operations
in magnetic circuits that require no new semiconductor devices and no
additional time.

This section will describe schemes in which the use of additional
time phases makes possible considerable savings in equipment and com-
plexity of logic. This is done simply by forming several replicas of a
set of independent data bits, and transmitting the replicas, in succession,
through a set of amplifiers, each time transposing the bits so that no bit
passes through the same amplifier twice. The replicas are collected sepa-
rately, combined logically and replicated to form the corrected next-state
variables. The logical function is OR or AND for asymmetric-error
channels, and Majority, for symmetric-error channels. *The important ad-
vantages of this method are the elimination of any redundant amplifiers,
and the great simplification of the encoding and decoding logic.* The
disadvantages of the method are the additional time—two extra READ phases
per bit cycle for asymmetric channels, and four extra for symmetric
channels, above the usual two READ and two WRITE phases, and the somewhat
greater number of register and logic cores.

Before describing the details of a specific system, the question of
amplifier group size will be considered. For asymmetric error correction,
two replicas must be transmitted, thus the minimal set size is two. The
question is, given a number of amplifiers to be divided into permutation
groups, is it advantageous to use a larger group size, with, for example,
a cyclic permutation of the data set through the amplifier group? These
alternatives are illustrated schematically in Fig. 3.20. Detailed
circuit descriptions will be given in Secs. III-B and III-C, below. As
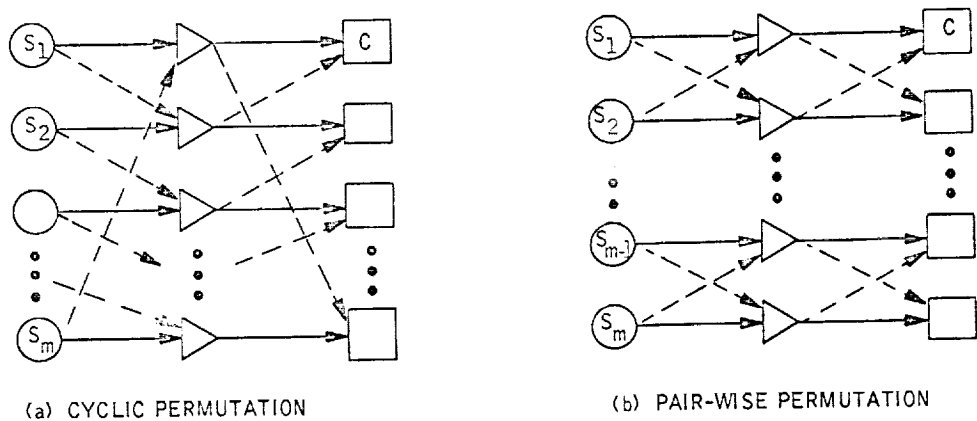in the case of parity-check coding, the smaller grouping is to be preferred

(a) CYCLIC PERMUTATION (b) PAIR-WISE PERMUTATION

RA—3196—86

FIG. 3.20 COMPARISON OF CYCLIC AND PAIR-WISE PERMUTATIONS
IN REPETITION-TRANSPOSITION SCHEMES

because it has greater power to correct double errors. This may be demonstrated by the following argument. Let there be m amplifiers in each system. In the cyclic scheme, the number of uncorrectable double error patterns is m, since any adjacent errors will result in one erroneous signal. In the pairing scheme, the number of uncorrectable double error patterns is only $m/2$, since both errors must occur in the same pair.[†]

B. ASYMMETRIC-ERROR CORRECTION SCHEME

The essential features of the scheme for asymmetric error correction are illustrated in Fig. 3.21. Figure 3.21(a) is a block diagram of one pair of channels, labeled X and Y. Two pairs of registers and two pairs of logic core sets are shown. They are so connected that channel X transmits data A_1 on the first pass, and data B_2 on the second, and channel Y transmits B_1 and A_2 on the first and second passes respectively. A given sense line passes through two registers, and the choice of registers is made by separate clock-phase drivers. The two A versions are collected in a single circuit. The single line output of the circuit, A_c, is the logical OR of the two stored inputs (for drop-out error correction) and it is used to form two replicas of the data for the next cycle. The
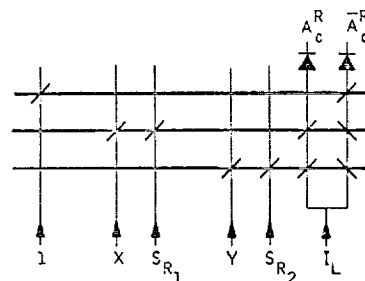

[†] Note that although the occurrence of an uncorrectable double error in the pairing scheme is half as likely as in the cyclic scheme, it causes twice the damage.
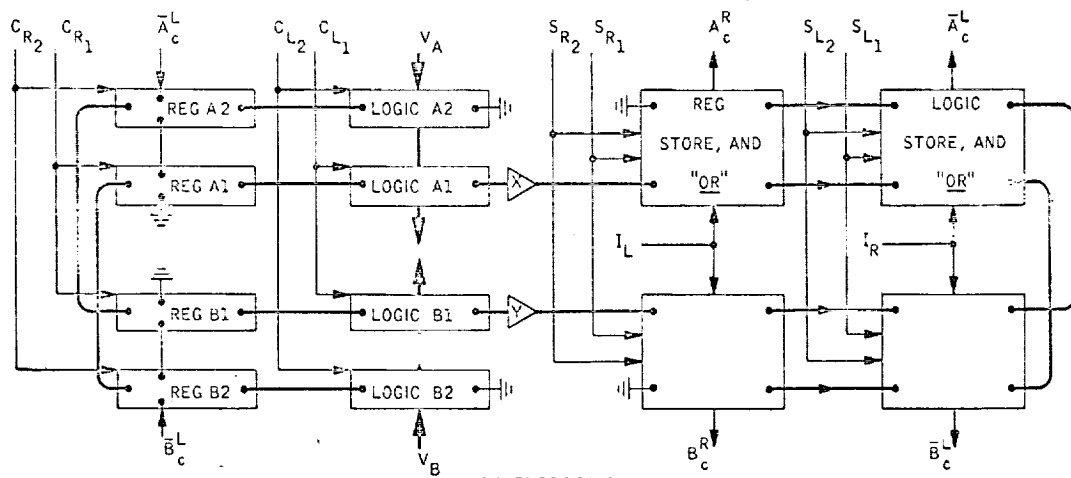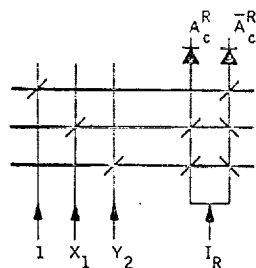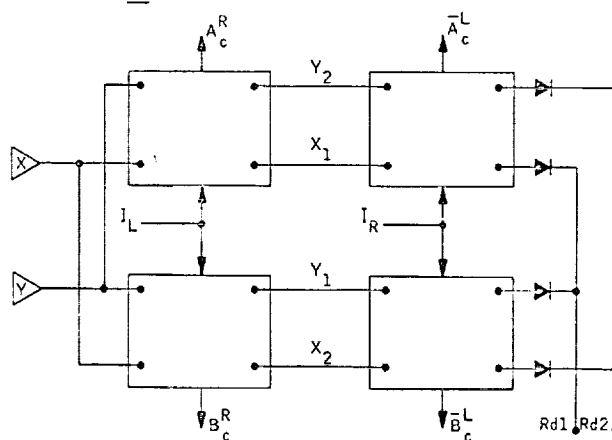
(a) BLOCK DIAGRAM
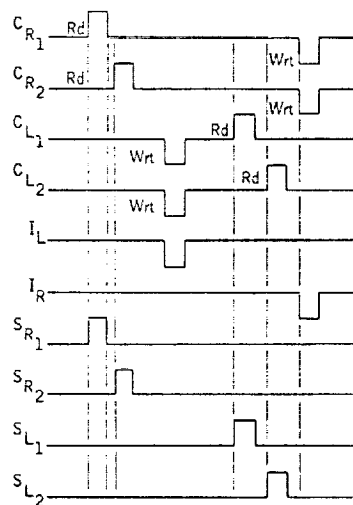
(b) CLOCK DIAGRAM

NOTE: $A_1 S_{R_1}$, $B_1 S_{R_2}$ - ALL 1/2 SELECT CURRENTS

(c) COINCIDENT-CURRENT TEMPORARY STORE AND "OR" SWITCH

(d) ALTERNATIVE CORRECTION SCHEME USING DIODE SELECTION RATHER THAN COINCIDENT-CURRENT SELECTION

RB-3196-90

FIG. 3.21   ASYMMETRIC-ERROR REPETITION-TRANSPOSITION SCHEME

137

two E versions are processed in the same way. This description holds equally for register and for logic data.

The time relations of the various clock phases are shown in Fig. 3.21(b). There are four READ times and two WRITE times, rather than the standard two READ times and two WRITE times. The clocks S_{R_1}, S_{R_2}, S_{L_1}, and S_{L_2} gate the collection of the four data samples into the proper combining circuits.
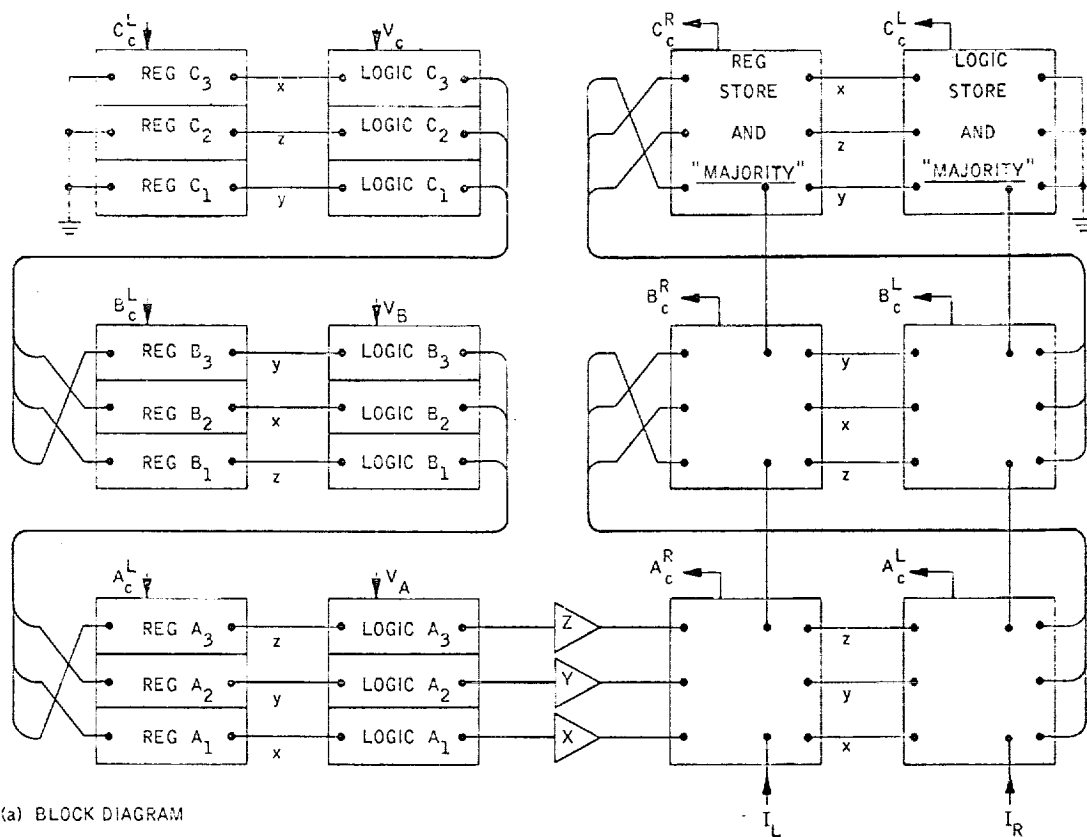
The storage-and-combining circuits present an engineering problem not previously met in the current-steering switches described in this report. This is the problem of preventing the output of an amplifier from destroying output data previously stored. Two gating systems are suggested, one based on current coincidence, the other based on diode switching.

The current gating scheme is illustrated in the right-hand side of Fig. 3.21(a). The block whose output is A_c^R is driven by the X and Y amplifier output lines. Its first signal is taken from the X line at S_{R_1} time, and its second signal is taken from the Y line, at S_{R_2} time. In order to keep the X data at S_{R_2} time and the Y data at S_{R_1} time from interfering, each sample is taken by a separate core, using coincident-current selection, as shown in the switch design of Fig. 3.21(c). The disadvantage of this approach is that coincident-current selection requires more careful control of current amplitude than simple inhibit-type selection.

The diode gating scheme is illustrated in Fig. 3.21(d). Here, the X output is split into two lines, each isolated by a diode, and selected by voltage pulses on the clock lines, labeled Rd_1 and Rd_2. Successive samples set separate cores, whose outputs are joined by logical OR at WRITE time. A magnetic circuit that is of simpler design than for current-coincidence gating is thus achieved, at the extra cost of two diodes per amplifier.

C. SYMMETRIC-ERROR-CORRECTION SCHEME

Correction of symmetric errors requires a triplet of amplifiers and three replicas of the data set, in order that a majority correction may be made. Figure 3.22 illustrates such a system. An attempt is made to indicate the wiring transposition scheme for distributing and collecting the successive data samples. Also, for clarity of drawing, the clock and

(a) BLOCK DIAGRAM

* CLOCK AND SELECTION LINES
  OMITTED FOR CLARITY

(b) CLOCK DIAGRAM

(c) COINCIDENT-CURRENT SELECTION
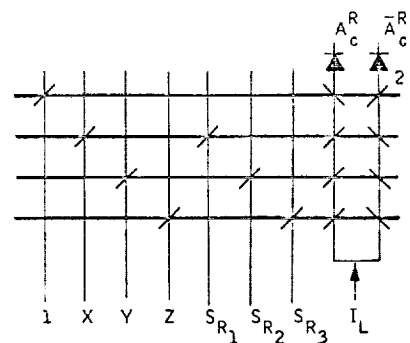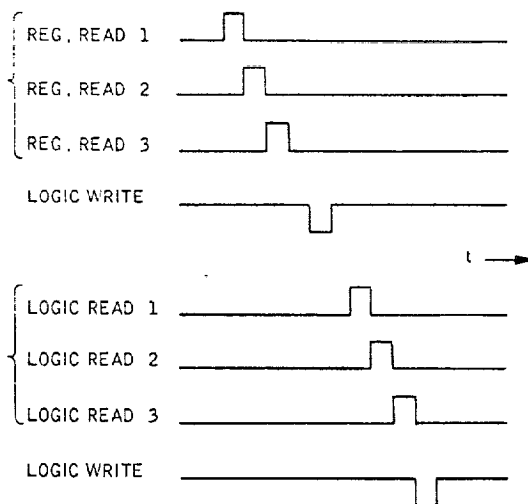TEMPORARY STORE AND MAJORITY SWITCH

RC—3196—102

FIG. 3.22 SYMMETRIC-ERROR REPETITION-TRANSPOSITION SCHEME

selection lines are not shown. Otherwise, the approach and notation of Fig. 3.21 are extended directly, so that detailed description is unnecessary. The same choice of diode or current coincident gating is available, with the latter illustrated.

D. MULTIPLE-ERROR CORRECTION

Multiple errors may be corrected by a direct extension of the above designs. For j asymmetric errors the number of data set copies, amplifiers per group, and read clock phases per transmission is $j + 1$. For j symmetric errors, $2j + 1$ data copies, amplifiers, and clock phases are needed. The design difficulties of the combining switches naturally increase with size.

IV HYBRID REPETITION AND PARITY-CHECK
REDUNDANCY SCHEMES

A. INTRODUCTION

The two scheme types of the preceding two sections may be distinguished by the exchange of time redundancy and equipment redundancy. Although the types appear quite different, there exist hybrid schemes that permit intermediate degrees of this exchange. Two examples will be presented here.

## B. SCHEME ALLOWING SOME REDUNDANT CHANNELS

In the first example, which is a scheme for correction of single symmetric errors, two independent data channels are supplemented by a third, redundant channel, and two transmissions are made. This may be compared with the parity-check schemes, which use an average of $1\frac{1}{4}$ redundant channels per data channel, and with the time-redundancy scheme, which requires three transmissions. The system scheme is illustrated in Fig. 3.23(a), in which a and b are independent data sources, and p is a data source which produces the parity function of a and b. The triangles represent the data channels, subject to symmetric errors. The memory elements q, s, and u store the first transmissions of a, b, and p, through Channels 1, 2, and 3 respectively, and Elements r, t, and v store the second transmissions of a, b, and p through Channels 2, 3, and 1 respectively.

The correction rules for a and b are

$$a_c = qr + (q \oplus r)(t \oplus u) = \text{Majority} (q, r, t \oplus u)$$

$$b_c = st + (s \oplus t)(q \oplus v) = \text{Majority} (s, t, q \oplus v) \quad .$$

The rule for a_c may be understood by the following reasoning: if $qr = 1$, then Channels 1 and 2 are either both faulty (a double error, unprotected) or both good. If $q \oplus r = 1$, then either Channel 1 or 2 is faulty, but under the assumption of a single error, Channel 3 must surely

AMPLIFIERS   TEMPORARY STORES

a,b INDEPENDENT DATA SOURCES

$p = a \oplus b$
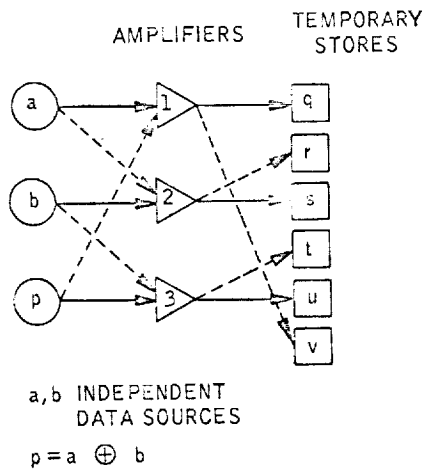
(a) SYSTEM SCHEME
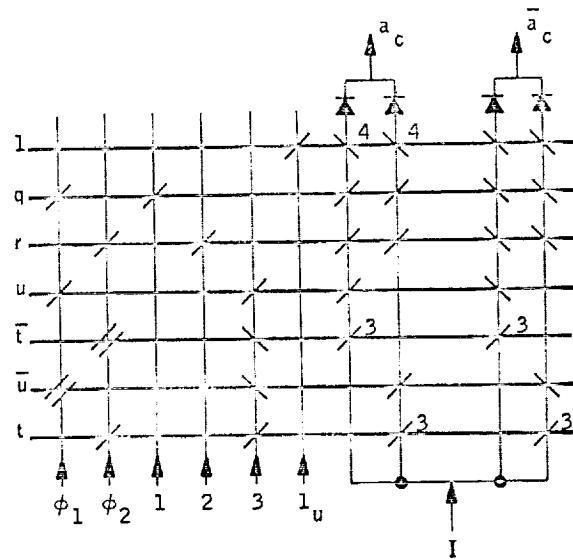
(b) CORRECTION SWITCH (COINCIDENT-CURRENT INPUT GATING)

RB-3196-105

FIG. 3.23 REDUNDANT-CHANNEL REPETITION-PARITY HYBRID SCHEME

be good. Thus, the true value of a may be obtained from $t \oplus u = b \oplus (b \oplus a) = a$. The same argument applies to b_c.

It should be noted that the scheme corrects single-channel failures, whether "permanent" (both transmissions faulty) or "transient" (only one faulty transmission).

The correction switch for implementing these rules, illustrated in Fig. 3.21(b), combines some of the more difficult problems of the switches for the parity-redundancy and time-redundancy schemes. Coincident current selection as shown in Fig. 3.21(b), or diode gating, is required to protect the two successive transmissions from interference, and the logic is such that a combination of diode logic and linear-output logic of fairly high weighting is needed.

The switch design is based on the decomposition

$$a_i = \text{Maj}(q, r, t \oplus u) = \overline{t}\,\text{Maj}(q, r, u) + t\,\text{Maj}(q, r, \overline{u})$$

$$= (4 * q, r, u, 3\overline{t}) + (4 * q, r, \overline{u}, 3t) \quad .$$

142

139

C. SCHEME ALLOWING NON-REDUNDANT CHANNELS ONLY

The second scheme also corrects single, symmetric errors. It uses no amplifier redundancy, and requires only two transmissions, but it requires more complex encoding and decoding. In Fig. 3.24, a, b, and c are independent data sources. In the first pass, a, b, and c are passed through amplifiers 1, 2, and 3, and the received versions are stored as a_0, b_0 and c_0, respectively. In the second transmission, the three pairwise parity functions $a \oplus b$, $b \oplus c$, and $c \oplus a$ are observed at Amplifiers 1, 2, and 3, and stored as x_0, y_0, and z_0, respectively.
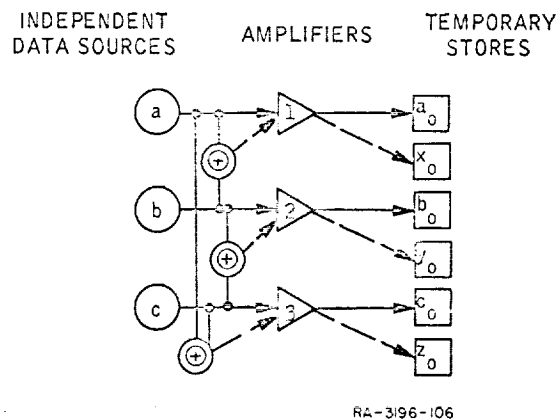

RA-3196-106

FIG. 3.24 REDUNDANT-LOGIC
REPETITION-PARITY
HYBRID SCHEME

The correction rules are:

$$a_c = a_0(a_0 \oplus b_0 \oplus x_0 \oplus 1) + (a_0 \oplus b_0 \oplus x_0)(c_0 \oplus z_0)$$

$$= \text{Maj} (a_0, \ b_0 \oplus x_0, \ c_0 \oplus z_0)$$

$$b_c = b_0(b_0 \oplus c_0 \oplus y_0 \oplus 1) + (b_0 \oplus c_0 \oplus y_0)(a_0 \oplus x_0)$$

$$= \text{Maj} (b_0, \ c_0 \oplus y_0, \ a_0 \oplus x_0)$$

$$c_c = c_0(c_0 \oplus a_0 \oplus z_0 \oplus 1) + (c_0 \oplus a_0 \oplus z_0)(b_0 \oplus y_0)$$

$$= \text{Maj} (c_0, \ a_0 \oplus z_0, \ b_0 \oplus x_0) \quad .$$

For a_c, we may argue as follows: if $a_0 \oplus b_0 \oplus x_0 \oplus 1 = 1$, then Amplifiers 1 and 2 are either both faulty (a condition that is not corrected) or both good. In the latter case, a_c may be taken to be a_0. If $a_0 \oplus b_0 \oplus x_0 = 1$, then the fault is in Channels 1 or 2, in which case, Channel 3 may be trusted. Thus, a_c may be obtained from $c_0 \oplus z_0 = c \oplus c \oplus a = a$. Since the system is symmetrical, the same argument applies to b_c and to c_c.

The correction switch (not shown), is of the same kind as for the previous scheme, but more costly. A possible decomposition is

$$a_c = b_0 c_0 \, \text{Maj} \, (a_0, \, \overline{x}_0, \, \overline{z}_0) + b_0 \overline{c}_0 \, \text{Maj} \, (a_0, \, x_0, \, z_0)$$
$$+ \, \overline{b}_0 c_0 \, \text{Maj} \, (a_0, \, x_0, \, \overline{z}_0) + \overline{b}_0 \overline{c}_0 \, \text{Maj} \, (a_0, \, x_0, \, z_0) \quad .$$

This calls for high-weight linear-output logic, of the form

$$b_0 c_0 \, \text{Maj} \, (a_0, \, \overline{x}_0, \, \overline{z}_0) = (7 \, * \, a_0, \, \overline{x}_0, \, \overline{z}_0, \, 3b_0, \, 3c_0) \quad ,$$

and eight diodes for each corrected variable and complement. It is not known whether this form is minimal.

D. COMMENTS

The schemes vary in the relative costs of extra time, extra amplifiers, encoder logic and decoder logic, where "costs" include both amount of equipment and design costs. Considerable room is thus given to the system designer for tailoring the redundancy to the costs of his circuit technology and to the particular logical structure of his system.

V CONCLUSIONS

A number of schemes have been presented for application of logical redundancy for fault masking, in the framework of a standard Inhibit-Core logic scheme. The only novel circuit required has been an extension of the presently used magnetic-core current steering switch to include combinational logic in the input and output circuits. This extension is well known, and has the advantage of requiring (for all but a small number of the codes described) no extra semiconductors, and no extra clock phases. Since this study did not include detailed circuit design, the upper limits of practical switch sizes (i.e., number of inputs and outputs and sizes of weights and thresholds) are not known. It is suggested that detailed circuit designs be carried out, at least on the simpler switches, such as the majority, for the (1, 2) code, the augmented majority switches, for the (3, 3) code, the complete decoding switch, for the (3, 3) code, and the coincident-current repetition--transposition majority correction switch. Although these are not the most efficient codes, they call for the simplest encoders, and their reliability improvements are quite substantial.

The choice of schemes must be made by the system designer on the basis of relative costs or limiting conditions applying to the different redundant elements--i.e., encoder equipment, amplifiers, decoder equipment, and time. His decision will also be influenced by the amount of reliability improvement needed and by whether the errors are symmetric or asymmetric.

Although the techniques presented have been directed toward a particular logic and circuit scheme--i.e., Inhibit-Core logic, with current-steering output circuits, it is not difficult to extend the techniques to other schemes. For example, there are other magnetic switches, without semiconductors, or with diodes or transistors, which could serve in an Inhibit-Core scheme, whose logical design would be covered by the switches described here. Furthermore, the general ideas of single-pass correction using error-correcting codes, of multiple-pass correction using transposition of channels, and of the possible hybrids, are applicable to any scheme in which protection of circuits that are essentially communication channels is needed--for example, memory sense and drive amplifiers.

APPENDIX

THEORETICAL RELIABILITY IMPROVEMENT OF SYSTEMS USING REDUNDANCY BASED ON ERROR-CORRECTING CODES

THEORETICAL RELIABILITY IMPROVEMENT OF SYSTEMS USING REDUNDANCY BASED ON ERROR-CORRECTING CODES

The improvement in system reliability given by use of redundant equipment may be measured by the ratio of the probabilities of system failure with and without redundancy. In systems whose corrections are based on parity check codes, system failure is defined as the probability that the number of faults exceeds the error-correcting capability of the code. In the following analysis it will be assumed that the elements fail independently with probability q, and that the equipment used for correcting errors is fault-free.

First, consider a system based on a single-error-correcting code of m independent data digits and k redundant check digits. The probability of system failure in the non-redundant system is

$$P_{F,N.R.} = 1 - P_{no\ faults} = 1 - (1 - q)^m \approx mq, \text{ for small } q.$$

The probability of system failure in the redundant case is

$$P^I_{F,R} = 1 - P_{no\ faults} - P_{exactly\ one\ fault}$$

$$= 1 - (1 - q)^{m+k} - (m + k)q(1 - q)^{m+k-1} \approx \binom{m + k}{2}q^2, \text{ for small } q.$$

The improvement is measured by

$$\frac{P_{F,R}}{P_{F,N.R.}} \approx \frac{1}{m}\binom{m + k}{2} q = Aq \quad .$$

Values for A for various single-error correcting codes are as follows:

m	k	A
1	2	3
2	3	5
3	3	5
4	4	5.25
4	4	7
5	4	7.2
6	4	7.5
7	4	7.9
8	4	8.25
9	4	8.67
10	4	9.1

As an example, a $(1,2)$ code with $q = 10^{-3}$ effects a reduction in the system failure probability of 333 times.

Second, consider an (m,k) double-error-correcting code. Here,

$$P_{F,R}^{II} = 1 - P_{\text{no faults}} - P_{\text{exactly one fault}} - P_{\text{exactly two faults}}$$

$$= 1 - (1-q)^{m+k} - (m+k)q(1-q)^{m+k-1} - \binom{m+k}{2} q^2(1-q)^{m+k-2}$$

$$\approx \binom{m+k}{3} q^3, \text{ for small } q.$$

Thus

$$\frac{P_{F,R}^{II}}{P_{F,N.R.}} = \frac{1}{m} \binom{m+k}{3} q^2 = Bq^2 \quad .$$

Values of B for various double error correcting codes are as follows:

m	k	B
1	4	10
2	6	28
3	7	40
4	7	41.3
5	7	44
6	8	60.7
7	8	65
8	8	70
9	8	75.5

As an example, a (1,4) code with $q = 10^{-3}$ effects an improvement of 100,000 times.

The relative improvements for a system composed of a number of groups is the same as that of the individual groups, for small q. It may be noted that the better (smaller) ratios are given by codes based on the smaller number of data channels per group. At the same time, the number of redundant channels needed in the system is greater for the smaller groups. For example, for protection of three data channels, the (3,3) code gives only $\frac{3}{5}$ the improvement of the (1,3) code, but it requires only $\frac{2}{3}$ the number of channels. The choice of codes thus depends on the cost of the channels and the value of the reduction in probability of failure.

ACKNOWLEDGMENT

147

REFERENCES

1. W. H. Kautz, "Automatic Fault Detection in Combinational Switching Networks," Tech. Report 1, SRI Project 3196, Contract M-44501 under NASA Contract NASw-8, Stanford Research Institute, Menlo Park, California (April 1961).

2. W. H. Kautz, "Codes and Coding Circuitry for Automatic Error Correction Within Digital Systems," Tech. Report 2, SRI Project 3196, Contract M-44501 under NASA Contract NASw-8, Stanford Research Institute, Menlo Park, California (January 1962).

3. J. Goldberg, R. C. Minnick, and W. H. Kautz, "Development of Techniques for Improving the Reliability of Digital Systems Through Logical Redundancy," Final Report, SRI Project 3196, Contract M-44501 under NASA Contract NASw-8, Stanford Research Institute, Menlo Park, California (May 1962).

4. L. J. Andrews, "A Technique for Using Memory Cores as Logical Elements," *Proceedings of the Eastern Joint Computer Conference*, pp. 39-46 (10-12 December 1956).

5. J. A. Rajchman and H. D. Crane, "Current Steering in Magnetic Circuits," *IRE Trans., PGEC*-6, 1, pp. 21-30 (March 1957).

6. J. von Neumann, "Probabilistic Logics and Synthesis of Reliable Organisms from Unreliable Components," in *Automata Studies*, Study 34, pp. 43-98 (Princeton University Press, Princeton, New Jersey, 1956).

7. E. F. Moore and C. D. Shannon, "Reliable Circuits Using Less Reliable Relays," *J. Franklin Institute*, Vol. 262, Part I, pp. 191-208 (September 1956), and Part II, pp. 281-297 (October 1956).

8. Rein Teoste, "Reliability of Redundant Computers," Report No. 21G-0029, Contract AF 19(604)-7400, Massachusetts Institute of Technology, Lincoln Laboratories, Lexington, Massachusetts (22 March 1961).

9. L. Depian and N. T. Grisamore, "Reliability Using Redundancy Concepts," *IRE Trans., PGRQC*-9, pp. 53-60 (April 1960).

10. G. D. Weinstock, "Matrix Analysis of Reliability for One-Shot Redundant Systems," *Electro-Technology*, 70, 5, pp. 99-103 (November 1962).

11. I. M. Ross, "Reliability of Components for Communication Satellites," *Bell System Tech. J.*, 41, 2, pp. 635-662 (March 1962).

12. T. A. Somer, "Redundant Pulse Amplifier," *IRE Trans., PGNS*-9, 5, pp. 7-16 (November 1962).

13. R. A. Ramey, "The Single-Core Magnetic Amplifier as a Computing Element," *Trans. AIEE, Communication and Electronics*, No. 4, pp. 442-446 (January 1953).

14. R. C. Minnick and J. L. Haynes, "Magnetic Core Access Switches," *IRE Trans., PGEC*-11, 3, pp. 352-368 (June 1962).

15. J. L. Haynes and R. C. Minnick, "Magnetic Core Access Switches," Appendices A and B, esp. particular Tables B1 and B2, Technical Supplement to the Final Report, SRI Project 3184, RADC-TR-61-117B, Contract AF 30(602)-2227, Stanford Research Institute, Menlo Park, California (May 1961).

16. V. I. Sobelman, "The Geometry of Threading Ferrite Core Matrices," in *Problems of Cybernetics*, Vol. 6, pp. 69-82 (1961); translated from the Russian by W. H. Kautz.

17. R. C. Minnick and R. L. Ashenhurst, "Multiple Coincidence Magnetic Storage Systems," *J. Appl. Phys.* 26, pp. 575-579 (May 1955).

18. "IBM Customer Engineering Manual of Instruction, 7302 Core Storage," Form 223-6836-1 (February 1961).

19. R. C. Minnick, "Linear-Input Logic," *IRE Trans., PGEC*-10, 1, pp. 6-16 (March 1961).

20. W. W. Petersen, *Error Correcting Codes*, p. 30 (John Wiley & Sons, Inc., New York, New York, 1961).

21. W. H. Kautz, "A Class of Multiple-Error-Correcting Codes for Data Transmission and Recording," Tech. Report 5, SRI Project 2124, Contract DA 36-039-SC-66381, Stanford Research Institute, Menlo Park, California (August 1959).

22. K. V. Mina and E. E. Newhall, "A Straightforward Way of Generating All Boolean Functions N Variables Using a Simple Magnetic Circuit," *IRE Trans.*, PGEC-10, 2, pp. 151-156 (June 1961).

23. H. W. Abbott and J. J. Suran, "Multihole Ferrite Core Configurations and Applications," *Proc. IRE* 45, pp. 1081-1093 (August 1957).

24. M. Karnaugh, "Pulse-Switching Circuits Using Magnetic Cores," *Proc. IRE* 43, pp. 570-583 (May 1955).

25. D. Rubin, "Complete List of Logic Terms for the MAGIC," Tech. Memo. 341-32 (Internal Memorandum), Jet Propulsion Laboratories, Pasadena, California (28 May 1962).

26. L. Lofgren, "Automata of High Complexity and Methods of Increasing Their Reliability by Redundancy," *Information and Control* 1, pp. 127-147 (1958).

STANFORD
RESEARCH
INSTITUTE

MENLO PARK
CALIFORNIA

Regional Offices and Laboratories

Southern California Laboratories
820 Mission Street
South Pasadena, California

Washington Office
808–17th Street, N.W.
Washington 6, D.C.

New York Office
270 Park Avenue, Room 1770
New York 17, New York

Detroit Office
1025 East Maple Road
Birmingham, Michigan

European Office
Pelikanstrasse 37
Zurich 1, Switzerland

Japan Office
c/o Nomura Securities Co., Ltd.
1–1 Nihonbashidori, Chuo-ku
Tokyo, Japan

Representatives

Toronto, Ontario, Canada
Cyril A. Ing
Room 710, 67 Yonge St.
Toronto 1, Ontario, Canada

Milan, Italy
Lorenzo Franceschini
Via Macedonio Melloni, 49
Milano, Italy